2018

The Impact of Cost on Feature Selection for Classifiers

Richard Clyde McCrae
Nova Southeastern University, mccraer@sympatico.ca

The Impact of Cost on Feature Selection for Classifiers

by

Richard McCrae, RM1718

A dissertation report submitted in partial fulfillment of the requirements

for the degree of Doctor of Philosophy

in

Computer Science

College of Engineering and Computing

Nova Southeastern University

2018

We hereby certify that this dissertation, submitted by Richard McCrae, conforms to acceptable standards and is fully adequate in scope and quality to fulfill the dissertation requirements for the degree of Doctor of Philosophy.

Sumitra Mukherjee, Ph.D.
Chairperson of Dissertation Committee

Nov 11, 2018
Date

Michael J. Laszlo, Ph.D.
Dissertation Committee Member

Nov 11, 2018
Date

Francisco J. Mitropoulos, Ph.D.
Dissertation Committee Member

Nov 11, 2018
Date

Approved:

Meline Kevorkian, Ed.D.
Interim Dean, College of Engineering and Computing

Nov 11, 2018
Date

College of Engineering and Computing
Nova Southeastern University

2018

ABSTRACT An Abstract of a Dissertation Report Submitted to Nova Southeastern University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

The Impact of Cost on Feature Selection for Classifiers

by

Richard C. McCrae

2018

Supervised machine learning models are increasingly being used for medical diagnosis. The diagnostic problem is formulated as a binary classification task in which trained classifiers make predictions based on a set of input features. In diagnosis, these features are typically procedures or tests with associated costs. The cost of applying a trained classifier for diagnosis may be estimated as the total cost of obtaining values for the features that serve as inputs for the classifier. Obtaining classifiers based on a low cost set of input features with acceptable classification accuracy is of interest to practitioners and researchers. What makes this problem even more challenging is that costs associated with features vary with patients and service providers and change over time.

This dissertation aims to address this problem by proposing a method for obtaining low cost classifiers that meet specified accuracy requirements under dynamically changing costs. Given a set of relevant input features and accuracy requirements, the goal is to identify all qualifying classifiers based on subsets of the feature set. Then, for any arbitrary costs associated with the features, the cost of the classifiers may be computed and candidate classifiers selected based on cost-accuracy tradeoff. Since the number of relevant input features k tends to be large for typical diagnosis problems, training and testing classifiers based on all $2^k - 1$ possible non-empty subsets of features is computationally prohibitive. Under the reasonable assumption that the accuracy of a classifier is no lower than that of any classifier based on a subset of its input features, this dissertation aims to develop an efficient method to identify all qualifying classifiers.

This study used two types of classifiers – artificial neural networks and classification trees – that have proved promising for numerous problems as documented in the literature. The approach was to measure the accuracy obtained with the classifiers when all features were used. Then, reduced thresholds of accuracy were arbitrarily established which were satisfied with subsets of the complete feature set. Threshold values for three measures –true positive rates, true negative rates, and overall classification accuracy were considered for the classifiers. Two cost functions were used for the features; one used unit costs and the other random costs. Additional manipulation of costs was also performed.

The order in which features were removed was found to have a material impact on the effort required (removing the most important features first was most efficient, removing the least important features first was least efficient). The accuracy and cost measures were combined to produce a Pareto-Optimal Frontier. There were consistently few elements on this Frontier. At

most 15 subsets were on the Frontier even when there were hundreds of thousands of acceptable feature sets. Most of the computational time is taken for training and testing the models. Given costs, models in the Pareto-Optimal Frontier can be efficiently identified and the models may be presented to decision makers. Both the Neural Networks and the Decision Trees performed in a comparable fashion suggesting that any classifier could be employed.

Acknowledgements

I wish to remember Dr. Ian Macleod for his many kindnesses and his refreshing ability to give direction when needed.

I wish to thank Prof. Millie Craig. Instructor, friend, and a genuine supporter.

I wish to thank Dr. T. Patrick Martin for his gentle guidance during my master's program.

I wish to thank Dr. Troy Savage for his ongoing support and continued belief that I could finish this effort.

I wish to thank my supervisor, Dr. Mukherjee, for his continuing guidance and most significant patience with my progress. Many thanks indeed.

I wish to thank my committee members, Dr. Laszlo and Dr. Mitropoulos. Their ongoing efforts to guide and support this effort are truly appreciated.

I wish to thank Dr. Ben Hoffman for his ongoing support and encouragement.

I wish to thank my parents for their belief that education is a valid goal, in and of itself, and for their many sacrifices over the years.

I wish to thank Dr. Meighen McCrae for her shining example as an academic.

I wish to thank Melissa McCrae for her support and encouragement over the years.

Finally I wish to thank my wife, Bev, for her ongoing commitment to me and this effort. Without her support, moral, emotional, and financial, this effort would never have been started, much less completed. Thank you.

Table of Contents

List of Tables

Tables

List of Figures

Figures

Chapter 1

Introduction

Background

Medical diagnosis is challenging and critical to safe and effective patient treatment. Successful diagnosis is often an iterative process beginning with an initial investigation, followed by some tests. Medical tests have several features. They all cost money. Many tests involve a level of discomfort or inconvenience for the patient while some are exceptionally painful or inconvenient. Many tests involve a measure of risk to the immediate or future health of the patient. In this description, tests have at least three dimensions of cost: dollar cost, pain or inconvenience cost and risk cost. The dollar cost may or may not be constant across all patients while pain plus inconvenience and risk may vary. Some people tolerate pain better than others, so the pain cost might be assessed differently by different patients. If a test causes an increased risk of cancer in 20 years, that risk might be meaningful to a 20 year old person, but much less so to an 80 year old person. The issue is that costs may vary from one situation to another.

The dollar cost may be borne by the individual patient or by another individual or group. Suppose that a clinic has an annual budget. When that budget is exhausted, the clinic can no longer function. It may be a simple if harsh reality that it is more effective to assess a large

number of patients with a slightly reduced accuracy rather than to assess a small number with a higher level of accuracy. In general, cost is a critical factor to consider.

Er, Yumusak, and Temurtas (2010) established that they could achieve high accuracy in diagnosing certain respiratory illnesses using an artificial neural network (NN). One feature on which they did not comment was the fact that every patient was given every one of 38 tests. It is possible that they might have achieved comparable results using fewer tests. If a trade-off between costs and accuracy can be established then a user will be in the position to make a better informed decision as to which features to measure.

This method can easily be generalized with respect to other classification objectives. Suppose an individual wants to decide which security (equity) to purchase (as Guresen, Kayakutlu, and Daim (2011) did). There are costs to gather different types of information and performance may vary depending on the information gathered. Suppose a researcher wants to detect disease in fish (as Gu, Deng, Lin, and Yu (2012) did), classify plants (as Yalcin and Razavi, (2016) did), or perform fault diagnosis (as Guo and Zhang (2008) did). Then the same general issues repeat. The best answer might be obtained with all features but there is a cost to collecting them; a lower cost, lower quality answer might be a better choice in some situations. Which choice should be made is not the focus of this effort. Rather, it is that the choices can be associated with a specified level of accuracy and at a known cost. Providing the end user with the relationship between accuracy and cost allows the end user to be better informed and so make better decisions. This result can be obtained only if the accuracy versus cost relationship can be determined and that will be the primary focus of this investigation.

For purposes of this investigation, two separate classifiers will be utilized: neural networks (NN) and decision trees (DT). The selection of these two is somewhat arbitrary as there are many other classifiers that might have been selected. However, both are represented broadly in the literature and the literature will be used to support their selection. The rationale for using a second type of classifier is to investigate whether or not the choice of classifier is relevant to the general process, described below.

Acceptable Results

While initially cast in the style of Er et al. (2010), these concepts can be generalized. Whether the question relates to diagnosing a patient, detecting disease in fish, selecting which security to purchase, or deciding any other question, the ability to determine an answer at an acceptable cost with an acceptable level of accuracy is critical. Resources are always constrained and better answers are always preferred. The trade-off between cost and quality of an answer is the critical feature of this investigation. The term 'acceptable results' will be used throughout this discussion. To some extent, this is a subjective term as it implies 'acceptable to some party' and that party may be under-defined. The specific values defining acceptable may vary depending on the party, the circumstances and the rationale for performing the classification.

Performance Measurements

There are several performance measures that will be used to evaluate the classifiers. A true positive (TP) is a case (instance in the test set) that was predicted to be true and actually was true. A true negative (TN) is a case that was predicted to be negative and actually was negative. A false positive (FP) is a case that was predicted to be true but was actually false. A false negative (FN) is a case that was predicted to be false but actually was true. Although these definitions apply to single instances, when used in formulas, as below, it is the count of the instances that is implied.

The total number of true samples is then TP + FN. Then the true positive rate (TPR) is defined as:

$$TPR = TP/(TP + FN).$$

The total number of false samples is then TN + FP. Then the true negative rate (TNR) is defined as:

$$TNR = TN/(TN + FP).$$

The false positive rate (FPR) is defined as:

$$FPR = FP/(TN + FP).$$

The false negative rate (FNR) is defined as:

$$FNR = FN/(TP + FN).$$

It is possible to eliminate some of these terms by noting (for example) that FNR = 1 – TPR or FPR = 1 – TNR.

The misclassification rate (MC) is the measure of items that were incorrectly predicted and is defined as:

$$MC = (FP + FN) / (\text{total sample})$$

$$= (FP + FN)/(TP + TN + FP + FN).$$

The overall classification accuracy rate (CAR) is the measure of items that were correctly predicted and is defined as:

$$CAR = (TP + TN) / (\text{total sample})$$

$$= (TP + TN)/(TP + TN + FP + FN).$$

(All definitions (Kelleher, Namee, & DArcy, 2015, pp. 404-414))

For purposes of this investigation, the CAR will be of primary interest. The author recognizes that this choice is somewhat arbitrary and that different users might have a different focus. It will be a trivial matter to adjust the process to focus on any accuracy measure which would have relevance to a given user.

Assume there are n features in a feature set FS. If the feature is present, then it is represented by a 1 otherwise by a 0. It can be observed that for a given classifier, given a feature set FS = $\{F_1, F_2, \ldots, F_n\}$, then the classifier will produce classification with some estimated level of accuracy.

Each feature has some cost associated with it; call it C_i, for the i^{th} feature. With a cost C_i for each feature F_i the total cost if all tests are run is $\sum(C_i * F_i)$ with Fi equal to 1 if the feature is present and 0 if the feature is not.

Of note, the term "test" is used to imply generating the value for a feature. In terms of a medical classifier, the test would be the test performed on the patient; for other classifiers it would mean evaluating some feature.

With a feature set with n features there are 2^n subsets so to test all subsets is infeasible for even moderately large feature sets. For any given classifier it is a reasonable assumption that the 'nearly' best answer will be obtained using the largest set of features available. Removing a feature may have no effect (the feature does not contribute to the classification produced), or it may degrade the classification (the feature does contribute to the result), or in some cases, it may marginally improve the answer. Phrased differently, removing a feature from a feature set would

not be expected to materially improve the results obtained.[1] (Aside: the removal of noisy features may slightly improve the results, and this behavior was occasionally observed. The classifier will typically render such features as meaningless or nearly so. Further, as the threshold value to be used is in reference to the accuracy obtained with the complete feature set, if the accuracy is slightly higher for some subset there is no damage done to the overall assessment.) Therefore, beginning with a complete feature set, the task was to develop a method to remove some of the features without excessively degrading the overall performance of the classifier. That objective was the primary subject of this investigation.

If the costs of features were static it would have been possible to incorporate them early in the process. However, such an assumption could not have been justified. Costs may vary from person to person, time to time and situation to situation. Therefore, the incorporation of the costs had to be deferred until after the set of acceptable feature sets has been determined.

The level of accuracy for a given classifier was used in two ways. Suppose, after the removal of some feature, the remaining subset produced a level of accuracy that was unacceptable (a binary choice). That subset was rejected. Further, there was no further need to investigate any of the subsets of the rejected subset following from the assumption that removal of features does not improve the results. Suppose, instead, that the subset produced results above the specified accuracy threshold. Then that subset was included in the 'acceptable' set of subset and its subsets were tested for acceptability. The set of subsets that was determined to be acceptable was termed the Acceptable Feature Set (AFS). The actual level of accuracy was also used further. When the

[1] It is recognized that it may be possible to get an execution of a classifier that produces a superior result with fewer features than does another execution using more features but that is simply due to the random nature of classifiers and the arbitrary assignments to training, validation and testing groups.

AFS was examined, clearly those subsets with a higher level of accuracy were preferred to those with a lower level. Hence, there was both a binary (acceptable or not) and a non-binary interest with respect to each acceptable subset.

Once the AFS was produced it was a trivial matter to evaluate the cost for each acceptable solution. At such time, the accuracy and the costs were matched and sorted both by cost and by accuracy. Producing a graph of cost versus performance was then a trivial matter. Of particular interest is the Pareto-Optimal Frontier, discussed below. The user could then select the optimal solution with respect to her situation.

Problem Statement

For any classifier, the cost of arriving at a decision may be a material issue. Collecting data is never a cost-free activity, and some data points can be exceptionally expensive. The cost component may have multiple dimensions (dollar cost, pain or discomfort, potential or actual risk or otherwise). It should be apparent that for any level of expenditure one would prefer the most accurate answer possible. Alternately, for any level of accuracy, the lowest expenditure would be most desirable. It is not difficult to extend this to a trade-off where accuracy and cost are in tension and can be made to describe an accuracy versus cost relationship.

Let $D^{F,m}$ be a classifier for diagnosis trained on a set of input features F using supervised machine learning method m. Initially, let m be constant. Then let $A_j(D^{F,m})$ be the expected accuracy of $D^{F,m}$ as determined by testing the trained model with respect to some accuracy measure j. As noted, the TPR, TNR and CAR were of primary interest. Let F^{max} be the set of all n features that might be used. Then the highest level of accuracy that could be expected would be

$A_j(D^{F^{max},m})$. Unfortunately, this also results in the highest cost with respect to measuring the input factors as all measures must be determined.

Several observations were made:

1. Some attributes in F^{max} contributed nothing to the estimate so could be eliminated without degrading the estimate.
2. The elimination of noisy features caused the accuracy to improve slightly. This does not invalidate the assumption that removing features tends to degrade the accuracy as the impact will be slight and may be zero if the classifier successfully renders their impact as zero.
3. It may be the case that some users would prefer lower accuracy if the corresponding cost were significantly lower. A user could use the Pareto-Optimal Frontier to make an informed trade-off between accuracy and cost.
4. Unless a lower limit of acceptability is established, it would have been necessary to test all 2^n - 1 combinations of features, which is infeasible. A lower bound on acceptability was established. For some datasets, the bound needed to be increased due to the very large number of feature subsets to be tested, especially when using random and increasing order of feature removal.

These were largely born out in the testing phase and are discussed further below.

Assumption:

$if\ F^1 \subset F^2$ then

$A_j\left(D^{F^1,m}\right) \leq A_j(D^{F^2,m})$ therefore, if F^2 is not acceptable, F^1 is not acceptable.

Therefore, the problem was to identify all F^* such that $A_j\left(D^{F^*,m}\right) \geq l_j$ where l_j is the accuracy threshold. Designate F^* as the Acceptable Feature Set (AFS). Establishing a method to determine the AFS was the major effort of this investigation.

As the cost for each test would be assumed to be given, it would be trivial to calculate these corresponding costs. Furthermore, each F_i is also associated with its corresponding $A_j(D^{F_i,m})$.

Therefore, let $\{(F_i, A_j(D^{F_i,m}), C_{ji}(D^{F_i,m}))\}$ with $F_i \in F^*$ be the set of all acceptable feature sets with the corresponding accuracy and costs. Ignoring the feature set itself we are left with

$\{(A_j(D^{F_i,m}), C_{ji}(D^{F_i,m}))\}$with $F_i \in F^*$ which describes the accuracy versus cost relationship. Call this the Accuracy versus Cost Curve (AvCC). Since the accuracy of each trained model was established, the cost-accuracy tradeoff was obtained.

Dissertation Goals

The primary goal of the dissertation may be stated as follows:

Given an *acceptable* classifier $D^{F,m}$ with respect to some minimum acceptable accuracy thresholds l_j for $j \in \mathcal{A}$, identify all subsets $F_s \subset F$ such that $D^{F_s,m}$ are also acceptable. Call this set the Acceptable Feature Set (AFS).

Since the size $|F|$ of the feature set was large for some datasets, it was computationally infeasible to train and test classifiers using all $2^{|F|} - 1$ subsets of features. We made reasonable assumptions to make the problem tractable:

Assumption: If a classifier $D^{F_s,m}$ is not acceptable, then all classifiers trained on a proper subset of F_s are also not acceptable.

Assumption: There is a lower bound with respect to the accuracy below which the solution will not be of interest. This bound, l_j, will be arbitrarily established. It establishes a threshold level above which the estimated accuracy is acceptable while below which the estimated accuracy is not acceptable. (Aside: the threshold level also had a material impact on the number of subsets that were evaluated. Hence, the threshold level was altered to be materially higher than initially planned for some datasets.)

Assumption: A 'nearly best' answer was obtained by using all of the features. Some results did improve slightly as some noisy features were removed. However, the assumption was that accuracy will soon start to fall off as more meaningful features are removed. Stated differently, if a subset of features produced an accuracy below the threshold level, then any subset of that subset would also be below the threshold level.

Assumption: Features can be ranked in a meaningful order (least to most significant or the reverse). Any such ranking is somewhat subjective as the methodology used to establish significance is a user choice (although once the choice is made there would be no further subjectivity). Several different approaches to ranking were employed, discussed below.

Under these assumptions, this dissertation applied depth first tree search to identify all subsets of F that result in acceptable classifiers. Nodes were represented by the feature set used to train and test the model. The root node was represented by the full feature set F^{max}. For a node represented by feature set F_s, a set of $|F_s|$ successor nodes was obtained as $\{F_s - \{f\} | f \in F_s\}$ by removing one feature at a time. Nodes representing classifiers that are not acceptable are removed

from the search tree and its subsets were never considered; all nodes remaining in the tree were acceptable classifiers.

A secondary goal of this dissertation was to investigate whether a judicious choice of the order in which features were removed could reduce the total number of classifiers to be trained and tested, thus reducing the total time needed to identify the set of acceptable classifiers. The set of acceptable feature sets was designated as the Acceptable Feature Set (AFS).

The relative importance of features in a classifier may be estimated. Our hypothesis was that the total number of nodes in our search tree could be reduced by considering a successor node obtained by the removal of a relatively more important feature before a successor node obtained by the removal of a relatively less important feature. This dissertation investigated this hypothesis by removing features in three different orders to generate successor nodes: descending, random, and ascending order of relative importance.

A tertiary goal of this dissertation was to demonstrate the practical benefits of identifying all acceptable models. For each acceptable classifier, its accuracy profile was presented in terms of all accuracy measures $j \in \mathcal{A}$. The cost associated with applying a trained classifier $D^{F,m}$ for diagnosis may be taken to be the sum of the cost of obtaining values for its input feature set F. Let c_i be the cost of obtaining a sample value for feature i. Then the cost of applying $D^{F,m}$ for diagnosis may be computed as $C(D^{F,m}) = \sum_{i \in F} c_i$.

The cost of applying an acceptable classifier with feature set F_s for diagnosis is $C(D^{F_s,m}) = \sum_{i \in F_s} c_i$. Using these costs, a non-dominated set of classifiers in the Pareto-Optimal Frontier of the cost-accuracy space was obtained. Decision makers could make informed decisions regarding the test results by selecting a subset of features used by some model in this

frontier. The set of acceptable classifiers and the corresponding accuracy profiles will be established. This will be referred to as the Accuracy versus Cost Curve.

The final goal was to determine the performance of DTs as compared to NNs. The resources required to execute the AFS was a primary area of interest. Also, the actual AFSs produced were compared. Of note, this was not to be construed as a validation of either DTs or NNs for classification as both are well supported in this role in the literature.

Research Questions

Research Question 1: What is an efficient process to identify all acceptable feature sets?

This question required developing a method to reduce the feature set by removing features one at a time, producing a subset, and then evaluating the result of that subset. If the result was at or above the acceptable level, then the feature set of that subset was included in the AFS. If it was acceptable, then its children were also evaluated. If it was not acceptable, then it was not included in the AFS and its subsets were ignored using the assumption that removing a feature does not improve the classifier's results.

Here, the term 'subsets' is used in the following sense. Suppose the current feature set is {1,1,1,0}, with the usual convention of 1 indicating the feature is present and 0 indicating that it is not. If we consider the rank (R_s) of each set to be the count of the number of 1s present then the rank of each of that set's subsets is $R_s - 1$. It was never the case that a feature is added back when producing the subsets. The initial (top level) set was that with all features present and the features were removed one at a time. The top level set then has rank *n*, with *n* the number of features. Therefore, the subsets of {1,1,1,0} are {{0,1,1,0}, {1,0,1,0}, and {1,1,0,0}}. Specifically, each

feature that was present (a '1') was removed in turn. (Note, the set representation does not reflect how the features will be coded, merely how they are conceptualized.)

Research Question 2: What percentage of the reduced feature sets are above the minimum quality threshold established?

This was a simple count of the AFS compared to the total possible number of subsets. The count increased as the threshold of acceptability was decreased.

Research Question 3: What percentage of the qualifying feature sets are on the Pareto-Optimal Frontier?

Answering this question required determining the Pareto-Optimal Frontier then comparing that with all of the AFS. This determination was relatively straight-forward once the AFS had been established.

Research Question 4: Does the order in which features are removed have an impact on the number of expansions required?

Answering this question required testing the implementation with different orders of removal. That is, the removal can be based on most significant first, least significant first or random order.

Research Question 5: What is the impact of using a different classifier on the AFS produced and the overall efficiency of the process?

This question was addressed by swapping the Neural Network classifier for the Decision Tree classifier. A simple comparison of the number of expansions produced one dimension for

comparison. A more interesting comparison was the actual feature sets produced for a given level of accuracy.

Relevance and Significance

Problem and those impacted

Cost is always a consideration in any activity. Quality of performance is also significant in most activities. Frequently, these two concerns are in tension. Better performance can often be obtained at a higher price. A lower price can typically be obtained by sacrificing the quality of the answer produced. This tension is obvious in many systems.

The cost of obtaining a specific feature is not necessarily indicative of how much that feature contributes to the quality of a decision. When the number of tests that might contribute to the quality of an answer increases, it becomes infeasible to evaluate all of the combinations of tests that might be used. Therefore, a heuristic is required to select the ones to use. There is no well-defined method to address that issue, especially when the costs of the tests are considered. Further, if the costs are dynamic, an additional layer of complexity is introduced to the general question.

The purpose of this investigation was to determine if there is a feasible method of reducing the search space to produce a lower cost answer of acceptable quality. Neural Networks were used as the evaluation criteria, but many other methods could have been used instead of them, for instance, decision trees, support vector machines or Naïve Bayes could also have been used. The rationale for using NNs was arbitrary but not capricious. They have a long history of being used for decision making or as classifiers and they have been shown to provide reasonably accurate results. The literature has numerous instances of NNs being used as classifiers in medical

diagnoses and in many other areas. While NNs were the primary classifier used, DTs were used as a confirmation that the results are not tied tightly to the choice of classifier. This confirmation will allow others to employ the classifier of their choice.

With respect to medical diagnoses, as well as the other items mentioned in the literature review and elsewhere in this document, Neural Networks have been used for diagnosing many things including:

- eye disorders (Syiam, 1994),
- ovarian cancer (Tan, Quek, & Ng, 2005),
- cirrhosis (Sun, Lu, Kobayashi, & Yahagi, 2005),
- carpal tunnel syndrome (Palfy & Papez, 2007),
- sleep apnea (Marcos, Hornero, Alvarez, Campo, & Lopez, 2007),
- macular diseases (Luculescu & Lache, 2008),
- Alzheimer's disease (Huang, Yan, Jiang, & Wang, 2008),
- thyroid disorders (Shukla, Tiwari, Kaur, & Janghel, 2009),
- endometrial cancer (Xiang, Tian, Zhang, & Dai, 2009),
- psychiatric disorders (Cui, Xiong, Zheng, & Chen, 2012),
- malnutrition related diseases (Arista-Jalife & Arista-Viveros, 2012)
- liver cancer (Kondo, Ueno, & Takao, 2012),
- flat footedness (Aruntammanak, Aunhathaweesup, Wongseree, Leelasantitham, & Kiattisin, 2013),
- stroke (Lin, Hsieh, & Hu, 2013),
- diabetes (Kumar, Sharma, & Agarwal, 2014),

- multiple sclerosis (Gutierrez, 2015)
- Parkinson's disease (Bazgir, Frounchi, Habibi, Palma, & Pierleoni, 2015),
- hypertension (Pytel, Nawarycz, Drygas, & Ostrowska-Nawarycz, 2015),
- prostate cancer (Sammouda, Wang, & Basilion, 2015),
- gum disease (Thakur, Guleria, & Bansal, 2016), and
- congenital heart septum defects (Jyothi & Vanisree, 2016).

Outside of medical diagnosis, neural networks have also found material application. Specifically, NNs have been successfully used in:

- fault diagnosis for steam turbines (Guo & Zhang, 2008),
- processing natural language (Collobert & Weston, 2008),
- stock market prediction (Guresen, Kayakutlu, & Daim 2011),
- detecting disease in fish (Gu, Deng, Lin, & Yu, 2012),
- predicting the strength of high performance concrete (Venu, Kiran, & Kiranmai, 2012),
- detecting disease in plants (Dhakate & Ingole 2015),
- fault diagnosis in lithium-ion batteries (Gao, Chin, Woo, Jia, & Toh, 2015),
- selecting recommendations for viewers on YouTube (Covington, Adams, & Sargin, 2016),
- plant classification (Yalcin & Razavi, 2016),
- automatically processing photographic enhancements (Yan, Zhang, Wang, Paris, & Yu, 2016), and
- retrieval of cooking recipes (Chen & Ngo, 2016).

Similarly, decision trees have been used in such medical diagnoses or treatment situations as:

- anemia (Maity, Sarkar & Chakraborty, 2012),
- bladder cancer (Floares & Birlutiu, 2012),
- monitoring posture and activities (Zhang & Sazonov, 2012),
- pulmonary disorders (Tartar, Kilic & Akan, 2013),
- Alzheimer's disease (Al-Dlaeen & Alashqur, 2014),
- brain tumor (glioblastoma) (Chaddad, Zinn & Colen, 2014),
- cardiovascular dysautonomias (Kadi & Idri, 2015),
- heart failure (Aljaaf, Al-Jumeily, Hussain, Dawson, Fergus and Al-Jumaily, 2015),
- thyroid disease (Shroff, Pise, Chalekar & Panicker, 2015),
- liver fibrosis (Ayeldeen, Shaker, Ayeldeen & Anwar, 2015),
- monitoring pregnancy (Lakshmi, Indumathi & Ravi, 2015),
- cerebral hemorrhage (Kumar & Krishniah, 2016)
- diabetes (Songthung & Sripanidkulchai, 2016), and
- breast cancer (Al-Salihy & Ibrikci, 2017).

Outside of medical diagnosis, decision trees have also found material application. Specifically, DTs have been successfully used in:

- detecting failure in internet sites (Chen, Zheng, Lloyd, Jordan & Brewer, 2004),
- network intrusion detection (Stein, Chen, Wu & Hua, 2005),
- identifying imposters on social media (Fong, Zhuang & He, 2012),
- evaluating students (Long & Wu, 2012),

- detecting fraud (Zou, Sun, Yu & Liu, 2012),
- stock trading (Ochotorena, Yap, Dadios & Sybingco, 2012),
- identifying individuals using biometric based identification (Kumar, Hanmandlu, Das & Gupta, 2012),
- security assessment of power systems (He, Zhang & Vittal, 2013),
- human gesture recognition (Oh, Kim & Hong, 2013),
- packet classification (Cheng & Wang, 2013),
- recognizing emotional aspects of speech (Yuncu, Hacihabiboglu and Bozsahin, 2014),
- detection of suspicious emails (Sharma, 2014), and
- assessing wine quality (Lee, Par, & Kang, 2015).

These lists were not meant to be exhaustive, merely illustrative. A simple search in the ACM on-line library for 'neural network' returned over 3700 hits. NNs are widely used in a variety of settings. The purpose of this investigation was neither to assert that NNs are the only or best choice for the evaluation portion nor to provide yet another example of using NNs for decision making. Rather, the purpose was to utilize NNs as a well-established tool to evaluate the heuristics proposed. So, the use of NNs was arbitrary, but considered. The same considerations apply for DTs.

Of note, this investigation made no effort to determine how the costs are generated. Nor was it concerned with which components of cost are considered. In an actual implementation, the costs would be sourced from others. In this investigation, costs were generated synthetically.

Scope of the problem, impact and benefit of a solution

Deciding how to classify items, situations, events or conditions is an exceptionally general problem. One might wish to categorize a disease. One might wish to determine which equities will perform better than others. One might wish to determine the species of a plant. One might wish to determine the effectiveness of various types of treatments. One might wish to determine the best time to plant or harvest a crop. The number of situations in which one might wish to determine an answer to a categorization problem is almost endless. For all of these questions, inputs must be provided. To test, that is, determine the value of, each input incurs some cost. The cost may be measured along a number of axes (dollar cost, inconvenience/pain, potential or actual risk, or some other dimension). The objective is not to list every potential dimension of cost but rather to note that there are many and these may vary from determination to determination, situation to situation, or person to person.

Costs are a concern in every environment. Resources are limited. The quality of a decision can frequently be improved by increasing the tests performed and so increasing costs. Eliminating some tests decreases the cost but may also degrade the quality of the answer obtained. This is an essential tradeoff. The point of this investigation is to determine exactly what that tradeoff curve looks like. Being able to utilize the AvCC for decision making would have potential value in any situation where a categorizer of any sort is being used.

It could be argued that there is a simpler way to approach this problem. That is, when the complete feature set is considered, simply remove each feature one-by-one, assess the change in accuracy and then use that information to guide the feature reductions. While that might produce acceptable results under some circumstances it would not be a general solution. Consider a situation where the subject's weight happened to be recorded in both pounds and kilograms.

Eliminating either of those should have no effect on the classifier. The researcher might reasonably conclude that neither were required and eliminate both of them. Data might, and almost certainly would, interact in more complex and subtle ways. Therefore, the single elimination/evaluation is inadequate. If the solution were extended to two-at-a-time removals and then three-at-a-time the number of cases required quickly expands. As there is no sound reason to stop at any number, the solution very rapidly becomes an exhaustive search which is infeasible.

Previous research and consequences of leaving the problem unsolved

While there have been attempts to address costs in categorizers, the attempts have not addressed the general issue of cost reduction related to feature elimination. Rather, the attempts typically address a particular problem and find a unique solution to that problem.

Vijayasarveswari, Khatun, Jusoh, Fakir, and Ali (2016) utilized a NN in the analysis of breast cancer. Their approach keyed on utilizing less expensive hardware to perform the tests required. While effective, it is not a generalizable approach (leaving aside the notion that if there is cheaper hardware that suffices it makes sense to use that instead). Seo, Yu, Lee, and Choi (2016) considered the overhead of running very large NNs and proposed modifications to the way in which inputs were categorized as a potential solution to this problem. While they acknowledged the cost of running the networks, they did not address the costs of obtaining the input values themselves.

Ji, Jiang, Zhao, and Zhai (2015) proposed a method of discriminating on the value provided by the tests used in their NN but did not address the cost component of the tests. While the value of a test is certainly of interest, the cost of obtaining that value must also be a concern.

Some work has been done that recognizes that costs are a factor or that reducing costs is generally desirable. However, with respect to directly assessing costs of obtaining the value of input parameters and using that information as a way to reduce costs, no work has been observed in this area. Therefore, this effort represents an investigation into an unexplored segment of decision making.

The consequences of solving this problem, even partially, could be quite significant. Every test performed consumes resources. In an environment where only the best answer is acceptable, a needless test simply wastes resources. In an environment where resources are constrained, spending those resources sub-optimally means that other investigations are foregone, whether immediately or at a later date, or that suboptimal conclusions may be produced. The effective use of testing resources will either save resources, improve the quality of the results obtained, or possibly both.

Addressing the research problem and potential for success

This study addressed the issue of cost in two distinct forms. The ultimate goal is to generate the AvCC which will allow a user to select the best choice for their situation, given that the user can define 'best' however desired. This first item has been the ultimate driver, but it was trivial to calculate once the AFS had been generated. The more difficult issue was the cost of finding the AFS. Given limitless time and resources, a brute force method would, eventually, produce a result which should match the AFS given that the assumptions stated elsewhere remain valid. However, limitless time and resources are not feasible. Hence, it was ultimately the cost of finding the AFS in a cost effective manner that was of concern.

The method proposed eliminated large branches of the search tree early in the pruning stage. Such early pruning vastly reduced the number of combinations that needed to be evaluated. The reduction was sufficient such that the time and resources required would be feasible in a wide variety of situations. Further, this study examined the impact of altering the selection for removal order (most significant feature, least significant feature, random feature). The most efficient removal order was to remove the most significant feature first was the most costly order was to remove the least significant feature first.

The pruning process described did produce an answer in less time than the brute force method would require. The issue then reduces to whether or not the speed-up was sufficient to justify the effort required. To a large extent this issue hinged on the minimum acceptance level that was selected. If the minimum was chosen close to the maximum, then even a modest loss of information (removal of only a few features) was sufficient to degrade the answer to the point where it was not acceptable. Therefore, the tree might have been pruned after only a few removals. The total number of subsets searched would have been a small multiple of the number of features. However, when the minimum acceptable answer was set very low, the classifier could produce a large number of acceptable, if less accurate, answers with only a small number of features. In these cases, the tree would have been pruned only after a large number of features were removed and there were many such nodes. The number of subsets considered was frequently very large and the resources required were significant. As it transpired, for several datasets the number of nodes tested was very high even when the threshold level was relatively close to the maximum value.

Adding to the knowledge base

This research added materially to the understanding of how costs can be used to influence the selection of features to be used in decision making. In every case where decisions are being made, there is a cost with determining the results. Costs are always a constraining factor, whether they are explicitly recognized as such or not. With this research concluded, the author can now provide a user with a process by which the contribution of a test to a determination can be evaluated. Cost-effective features can be included in the test suite. Cost-ineffective ones can be eliminated. Further, the user will have the ability to examine the AvCC to determine how best to deploy scarce resources. This method has demonstrated that it can readily accommodate dynamic costs.

Through this investigation, an understanding of how various factors impact the cost of generating an AFS was gained. This understanding will allow a user to generate a new AFS efficiently. As there are many classification problems and virtually all of them require inputs of varying costs, this understanding may provide wide-spread benefit.

Additionally, suppose a new feature is suggested for the test suite. A user now has the ability to evaluate whether to include the feature or not. A medical diagnostician may also be able to determine that, if the cost is less than a certain amount, then it should be included, otherwise not. Alternately, an individual suggesting (perhaps selling) the new test might be able to determine the ideal price to charge to obtain the maximum profit for that test. The decision making process will be improved.

Additionally, knowledge will be gained regarding the impact of lowering the acceptability threshold. As the level is decreased, the number of acceptable feature subsets increased

dramatically. By measuring the effort required to solve problems of a different sizes, insight was gained into the size of problems for which this approach is suitable.

Potential for generalization

The initial impetus for this investigation was to study the removal of select tests from a test suite which was the input to a NN being used to categorize specific diseases. It is apparent that different tests may have dramatically different costs. It is also apparent that different tests will have different impacts on the results returned by the NN. It is possible that in some environments, the limiting factor in selecting which tests to use would be obtaining the best performance (that is, highest accuracy possible). Still, there might still be opportunity for cost reduction if some tests might be found to provide no useful information. However, as the level of acceptable performance is lowered, the opportunity to reduce cost increases. One might argue that when performing a diagnosis, only the best available performance is acceptable, however that is not necessarily the case. Consider a case where disease A and B both have comparable treatments. In such a case, it may not make good sense to spend limited resources to differentiate between the two (provided that one can be satisfied that it is either A or B and not something else). Further, not all environments are rich enough to support exhaustive testing. There may not be enough trained individuals to perform all of the tests; the materials required to perform all of the tests may not be available (or affordable); some tests may pose excessive hazards or cause pain to some or all patients; or the patients themselves may not be able to make visits to the places where certain tests are to be carried out. So, it may be the case that one can perform a very small number of diagnoses with a high level of accuracy or a much larger number with a slightly lower level of accuracy. The interaction of these relationships can all be revealed with a suitable AvCC.

While it is a trivial matter to describe an algorithm that will test every combination of tests (and so build an exact cost for each subset of tests) it is infeasible to run such a process for even modest sized feature sets. As noted elsewhere, the run time is proportional to $2^n - 1$, with n the number of tests. For each test, a NN, or other classifier, must be generated. The run times are simply not feasible for even moderately large values of n. Therefore, the methods described in this paper were developed. These methods have been tested and found to be satisfactory; the potential for generalization is significant. There are many comparable systems that use NNs or other classifiers to assist with a decision. Being able to efficiently generate a new AvCC given a different set of costs would allow the users of such systems to be more informed as to the value their systems would generate.

Neural Networks were designated as the primary classifier for this investigation, with decision trees as the confirmation classifier. That choice was arbitrary. Using a NN was convenient because there are many pre-packaged environments that are readily available (e.g. R, Matlab). Further, NNs and DTs have been used in so many classification systems that there is a high level of confidence that they actually do work correctly and effectively. However, any process that uses inputs to make a decision would have sufficed (e.g. Support Vector Machines, Naïve Bayes classifiers). It is not the decision making process that was being examined, rather the method to extract an appropriate subset of the input tests. Therefore, there is additional potential for generalization with respect to the type of classifier used.

From the above discussion, it is not difficult to conclude that this work can be generalized. There are a significant number of NN and DT systems that are used and benefit could be gained from understanding the cost/accuracy tradeoff. Further, it is trivial to swap out the NN classifier and substitute a different classifier.

Potential for original work

While it is not uncommon to see those writing about classifiers mention cost as an issue, there is negligible evidence that cost has been researched to any material extent. Further, dollar cost is frequently a proxy for many other measures. Humans put a dollar cost on property, on opportunity, on inconvenience, on discomfort, on risk, and on many other things. These costs may not always be explicitly stated (or precisely measured), but that does not mean they are not real. It is not difficult to imagine many situations where one might put a dollar cost on things that are normally measured in other terms. The person who pays extra to drive on a toll road implies a dollar cost for convenience (or time saved). The person who pays extra to sit in a first-class airline seat implies a dollar cost for comfort. The person who pays extra for safety features in an automobile implies that money can be exchanged for safety. The number of examples of trading money for something else would be almost endless and the objective is not to provide an exhaustive list of such exchanges, simply to note that such exchanges are common.

Therefore, with the potential to measure many items of value in terms of money, one can begin to examine a more extensive trade-off. When there are dozens or perhaps hundreds of terms involved, the utility of a common currency should become apparent. Further, not all individuals will place the same value on certain elements. A procedure that would make an individual sterile would be of no consequence to a 90 year-old person but might be devastating to a 20 year-old one. An investigation that would degrade one's physical performance in a foot race by two percent would not be meaningful to most individuals but might be catastrophic to a professional athlete. A procedure that cost $1,000 might pose an unbearable burden to one person, yet be a trivial amount to another. Costs may be exceptionally dynamic and completely person specific.

Most of these implications have not been explored. While it is trivial to look up an example of NNs or DTs being used as an aid in medical diagnosis or to investigate other categorization problems (see section: Relevance and Significance), costs are not normally a consideration. This effort presents an investigation into an area that has been neglected.

Barriers and Issues

It is not a trivial task to perform a diagnosis. Acquiring and organizing the data needed to train a neural network or other classifier may present a material challenge and a significant cost. Although it is evident that neural networks and other classifiers can be trained to diagnose accurately in many cases, the issue of the cost of acquiring the data for the diagnosis has received scant, if any, attention. Part of the reason for the lack of attention may be that there has been no method developed to understand the relationship between cost of a particular test and the benefit of that test, particularly when combined with other tests. This study is focused on the relationship between accuracy and cost and can only do that by understanding the collective value of subsets of the feature set when taken together.

The principal difficulty in approaching this problem is that there is no known relationship between the features. That is, one cannot isolate the contributions of each feature. The relationship between a collection of features and the resultant accuracy of the classifier where there will be multiple features present is simply not known from their individual contributions.

Therefore, the only currently practical way to determine the relationship is to test the various combinations. For feature sets of more than a modest cardinality, the task of computing all combinations is simply infeasible. Therefore, some simplifying techniques or assumptions must be applied. The simplifying assumption that was employed in this study is that if a feature is

removed from a feature set, the accuracy of the classifier does not go up materially. Phrased differently, if a subset of features falls below the acceptable level, removing a feature will not result in an acceptable subset. That concept was used to reduce the infeasible brute-force method to a manageable one.

As remarked elsewhere, there are a great number of areas in which categorization is a material concern. For virtually all such situations, cost is a material factor. Therefore, obtaining the accuracy/cost curve for categorizing a problem in a timely manner is both challenging and valuable. The general solution will be initially cast as a solution to a medical diagnosis problem. However, there is nothing in the concept of classifier or costs that is peculiar to medical diagnosis so the solution will have general applicability.

Assumptions, Limitations and Delimitations

The assumptions below have been introduced elsewhere in this paper. They are included here for completeness.

Assumption: If a classifier $D^{F_s,m}$ is not acceptable, then all classifiers trained on a proper subset of F_s are also not acceptable.

Assumption: There is a lower bound with respect to the accuracy below which the solution will not be of interest. This bound, l_j, will be arbitrarily established. It establishes a threshold level above which the estimated accuracy is acceptable while below which the estimated accuracy is not acceptable.

Assumption: A 'nearly best' answer will be obtained by using all of the features. The results might improve slightly as some noisy features are removed. However, the assumption is that accuracy will soon start to fall off as more meaningful features are removed. Stated differently, if

a subset of features produces an accuracy below the threshold level, then any subset of that subset will also be below the threshold level.

Assumption: Features can be ranked in a meaningful order (least to most significant or the reverse). Any such ranking is somewhat subjective as the methodology used to establish significance is a user choice (although once the choice is made there would be no further subjectivity).

There are no apparent limitations other than that the number of evaluations increases dramatically as the number of features increases. Of note, the calculation of the AFS need only be done once for a given set of features, it does not need to be repeated when the prices are updated. The only delimitations are those related to the choice of datasets. That is, it is necessary to use actual datasets, hence the choice is arbitrary, regardless of who actually makes the choice.

Definition of Terms

All of the following items are also included in the List of Acronyms. Most of the acronyms are common terms that are frequently used. The few that are detailed here are not commonly employed but are specific to this investigation.

ACC^0 (initial accuracy) is the initial accuracy of the classifier. Given a classifier and a complete set of features, the classifier will produce an accuracy of ACC^0.

The Acceptable Feature Set (AFS) is the set of all subsets of features that, using a given classifier, produce a level of accuracy equal to or above the threshold level.

The Accuracy versus Cost Curve (AvCC) is the relationship between accuracy obtained and the cost of obtaining that accuracy. Ultimately, it is the goal of this investigation.

List of Acronyms

The following Acronyms have been used in this paper:

- ACC^0: Initial accuracy of the classifier
- AFS: Acceptable Feature Set
- AvCC: Accuracy versus Cost Curve
- cAFS: count AFS
- cFST: count Failed Sets Tested
- cPOF: count of sets on POF
- cPDS: count of Potential Data Sets
- cTST: count Total Sets Tested
- CAR: Classification Accuracy
- DT: Decision Tree
- NN: Neural Network
- FN: False Negative
- FNR: False Negative Rate
- FP: False Positive
- FPN: False Positive Rate
- MC: Misclassification Rate
- TN: True Negative
- TNR: True Negative Rate
- TP: True Positive
- TPR: True Positive Rate

Summary

Medical diagnosis is a challenging and expensive task. There are many examples of NNs and DTs being used for medical diagnosis as well as for other decision making efforts. Not all features will necessarily contribute meaningfully to the accuracy of a classifier. All features have a positive cost related to their collection. These costs may exist in multiple dimensions (actual dollar cost, inconvenience or discomfort, potential health risks or other) but can be reduced to a dollar cost.

Reducing the features employed generally reduces the accuracy but also reduces the cost of the assessment. Therefore, it is easy to consider a relationship between the cost of producing an answer and the anticipated accuracy of that answer. This relationship is the essential focus of this investigation.

This study succeeded in finding the relationship between features measured and accuracy for several specific datasets. This was done by establishing a minimum acceptable accuracy then generating all of the subsets of features that meet or exceed that accuracy level. It is assumed that the 'nearly' best answer is achieved when all features are present. As features were removed, the accuracy of the answer deteriorated. It was also assumed that, below a certain level of accuracy, an end user would have no interest in the answer.

Hence, it was easy to establish two bounds. The upper bound of accuracy was the value obtained when all features are present. Whether the accuracy improves slightly with the removal of 'noisy' features is of little consequence. The lower bound of accuracy is the lower limit at which a user might be interested in the answer. Between those two values, all subsets of features are of interest. It was the objective of this investigation to find them in an efficient manner and

then present them as a Pareto-Optimal Frontier. Then end user would then be able to apply any dynamic cost function to the frontier to determine their optimal course of action.

Chapter 2

Review of the Literature

There are eight areas of particular interest in this literature review. The first area is to examine the support for using Neural Networks as the classifier. The goal is to demonstrate that NNs have a long history of being used to accept input factors and produce a classification. The classification might be a simple binary situation (e.g. to determine if this is a buying opportunity or not, or to determine if a tumor malignant or benign). Alternately, the classification might be to determine which of several choices is best (e.g., to determine if the patient has disease A, B, C, or D).

The second area of interest is to highlight that very little effort has been put into studying the impact of costs on a diagnosis. While it is not possible to show that no effort has been expended to examine costs (one would have to examine all papers produced, which is infeasible), it is possible to review a material number and comment as to whether or not minimizing cost via selection of factors was a non-trivial component of the study.

The third area will be to establish that decision trees are also an established classifier for performing diagnosis. The second type of classifier is desired to establish that the process is not

tightly tied to neural networks. That is, any classifier of choice could be selected and the process should still be valid.

Of note the first three areas of interest are considered together. They form the first subsection of the literature review.

The fourth area of interest is to review active versus passive classifiers. Greiner, Grove, and Roth (2002) discussed costs in the context of an active versus a passive classifier.

The fifth, sixth and seventh areas all relate to feature set reduction. The articles selected represent some of the early work performed in this area and have been referenced many times in the literature. Hence, they represent key early developments. In the fifth section, reduction using the filter method is reviewed. In the sixth section, early work on the wrapper method is discussed. In the seventh section, the hybrid method is reviewed. These seminal efforts will be traced forward to the present.

For each study discussed, remarks will be produced as to the nature of the study and the results obtained. Further, the extent to which cost was considered will be noted.

The eighth section will summarize the current state of affairs. It will largely follow the organization of Li et al. (2017), but will also incorporate other elements.

Neural Networks, Decision Trees and Cost Considerations

Er, Yumusak, and Temurtas (2010) noted that neural networks had been used previously for respiratory and other medical diagnoses. These 3 researchers worked (not always together and frequently with others) on 7 of the 37 papers used as reference material for their article. Further, they referenced that paper in four later articles. This specific article can be viewed as one in an

ongoing series where they (and many others) delve into this topic. They used a variety of NNs to diagnose respiratory illnesses. They considered several different types of NNs including multi-layer neural network (MLNN), probabilistic neural network (PNN), learning vector quantization (LVQ) neural network, generalized regression neural network (GRNN) and radial basis function (RBF) neural network. These were used to produce a differential diagnoses between TB (tuberculosis), COPD (chronic obstructive pulmonary disease), pneumonia, asthma, lung cancer, and no illness present. Their dataset used 38 input features. They used Matlab for their calculations. Accuracy varied from 88% with GRNN to 92% with PNN. There was no discussion of costs.

El-Solh, Hsiao, Goodnough, Serghani and Grant (1999) described using Neural Networks to diagnose tuberculosis. They reported that their methods (General Regression Neural Network (GRNN)) outperformed physicians' clinical determinations. There was no discussion of costs.

Kabari and Bakpo (2009) developed a neural network which was successful in diagnosing selected skin diseases. They reported performance in excess of 90% accuracy. They mentioned that costs were a factor but did not follow through with any analysis of the cost of tests. They did remark that NNs may help reduce costs, but did not elaborate on the details of how this might be achieved.

Vijayasarveswari et al. (2016) proposed a NN solution for screening for breast cancer. They reported accuracy ranging from 82% to 100%. Their method did consider feature reduction, but it was an across the board effort; it was not selective to individual circumstances. The authors did note that their system was materially less expensive than others, but it was designed as a replacement for a more sophisticated system. It did not use the evaluation of the tests used by the

NNs as a method to choose between various tests. Rather, it was completely a one-off resolution; their contribution was designing a less expensive replacement of a more expensive system.

Ibrahim, Shamsuddin, Saleh, Abdelmaboud, and Ali (2015) used a multilayer perceptron with differential evolution technique to produce NNs used in the diagnosis of breast cancer. They used the University of Wisconsin Hospitals, nine parameter dataset (available at https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original)). They commented that "The multi-objective evolutionary algorithm used in this work optimizes error rates and architectures of the MLP network simultaneously" (p. 424). Overall, they reported accuracy rates averaging 97.51%. There was no discussion of costs.

Liu and Dong (2012) also used NNs to diagnose breast cancer using the same Wisconsin Hospital data as did Ibrahim et al. (2015). They used the Levenberg-Marquardt algorithm instead of the Gradient descent algorithm. The rationale stated was that "L-M algorithm uses the approximate information of the second derivative, and is much faster than gradient method" (p. 1239). They reported an accuracy rate of 98.8%. There was no discussion of costs.

Chunekar and Ambulgekar (2009) applied neural networks to three different datasets to diagnose breast cancer. They reported accuracy ranging from 70.7 to 98.8% which supports the notion that NNs can be useful for diagnosing. Of note, the relatively low 70.7% result was not explained. They did remark that "these neural network based clinical decision support systems avoid unnecessary excision and expense" (p. 895) which, at a minimum, recognized that pain and expense are factors that might be considered. However, it was within the general context of using NNs instead of (or in support of) other methods. No analysis of the cost of the tests used in the NNs was discussed.

Elveren and Yumuşak (2011) proposed using a genetic algorithm (GA) to assist with training of a NN to diagnose tuberculosis. When using GA to train NN, the weights and biases were encoded into chromosomes, therefore a generation represented a set of NNs. They thought that the GA might reduce the time required to train the NNs but concluded that this was not the case. They did document that their method was successful in performing the diagnosis with accuracy of 94%. No analysis of the cost of the tests used in the NNs was discussed.

Coppini, Miniati, Paterni, Monti, and Ferdeghini (2007) used feed-forward multilayer neural networks to diagnose emphysema in patients with COPD. Their reported accuracy was 90%. There was no indication that they considered costs of the tests as a factor.

Er, Sertkaya, Temurtas and Tanrikulu (2009) used multilayer, probabilistic and learning quantization neural networks for diagnosis of COPD and pneumonia and recommended that these approaches could be used to assist physicians in forming their diagnoses. There was no consideration of costs of the tests performed.

Ayeldeen, Shaker, Ayeldeen and Anwar (2015) used Decision Trees to classify liver fibrosis. They achieved an accuracy of over 93%, which was higher than the rate typically achieved by manual analysis. They did not comment on the costs of the tests performed.

Al-Salihy and Ibrikci (2017) used 32 attributes to classify breast cancer using Decision Trees and achieved an accuracy rate of 97.7% when using the Wisconsin Breast Cancer dataset. There was no mention of the cost of acquiring the data points.

Flares and Birlutiu (2012) used DTs to diagnose bladder cancer. They reported high accuracy (>95%) and high robustness along with low model complexity. There was no mention of the cost of acquiring the data points.

Al-Dlaeen and Alashqur (2014) developed a model to predict Alzheimer's disease using decision trees. They based their model on the ability of each feature to add to the information gain. There was no mention of the cost of acquiring the data points.

From this literature review it is apparent that both NNs and DTs can be developed to generate acceptable results in diagnosing a wide range of diseases. However, there is scant, if any, consideration of the costs of acquiring the data needed for the classifiers. Further, there has been no observed consideration of the accuracy obtained versus the cost of the data used. At this point, this area of interest seems to have been neglected.

Active versus Passive Classifiers and Costs

Greiner et al. (2002) discussed costs in the context of an active versus a passive classifier. A passive classifier, given a set of features, must produce an answer using just the values provided. In contrast, an active classifier can either produce an answer or ask for more information. The issue then becomes which information to request and what is the impact on the overall cost of achieving an answer. For each new data point requested, the classifier incurs a cost which is dependent on the test performed. While they never commented on whether the costs would be dynamic or static it is apparent from their work that dynamic costs would be acceptable.

There are two major concerns with this approach. Due to the sequential nature of the process there may be material time concerns. That is, the classifier may make repeated calls for new tests that necessarily increases the overall amount of time required to reach a decision. Greiner et al. (2002) noted that the time considerations may make some requests pointless if the time to process the request exceeds the time remaining to make a decision. Further, significant

portions of the classifier must be regenerated each time a new diagnosis is required. This process might incur significant machine costs.

Further, their process is not guaranteed to find the cheapest solution. For example, suppose there are three potential tests (A, B and C) and the classifier has determined that a true result, T, can be determined by (A /\ (B\/C) \/ (B/\C), otherwise the result is false, F. Let the costs be one for A and two for each of B and C. The classifier might select A first. Suppose it is false. It would then select either B or C, which cost the same, suppose it was B and that B is true. C is next required and, if true the result is T otherwise F. It can be observed that the value of A was not required, yet it was paid for. Hence, the cost was not optimal. It should be further noted that in all of the above cases the time consideration was assumed to be acceptable (that is, the answer could be deferred until the next test was completed). Also, the classifier might make different choices based on its understanding of the distribution of the results in the population (for example, A = true might be exceptionally rare so would not have been selected).

Therefore, while interesting in that it addresses costs, the focus of Greiner et al. (2002) is materially different from that suggested in this proposal. Greiner et al. (2002) incrementally address costs and do so dynamically at each step (for active classifiers). The approach suggested in this proposal is not at all comparable. Essentially (as elaborated elsewhere), this proposal is to determine all subsets of features that will yield an acceptable accuracy (that is, accuracy above some minimum) with the specified classifier. The total cost to evaluate each subset can then be determined dynamically and the cost versus accuracy relation is immediately apparent. So while both approaches might be said to be dynamic and concerned with costs, the way in which they are dynamic is completely dissimilar and their approach to costs is also completely dissimilar. The approach of Greiner et al. (2002) dynamically selects the next feature to evaluate at run time using

costs as a factor. The approach proposed in this paper is to pre-establish the set of acceptable subsets of features and dynamically calculate the costs of each subset based on the (at least potentially) dynamic costs associated with each test.

A Brief Survey of Significant Developments in the Early History of Feature Set Reduction

Approaches

Feature reduction techniques frequently fall into one of three categories. The Filter algorithm uses an independent algorithm that performs feature reduction according to some quality metric associated with the features, typically based on some statistical measurement. It is independent of the classifier algorithm and is used as a preprocessing step. The Wrapper algorithm uses an algorithm that works in conjunction with the classifier. Feature reduction is guided by the performance of the classifier. The Hybrid algorithm seeks to take advantage of features of both the filter and the wrapper methods and is suggested when the datasets are large. Typically, a hybrid algorithm will use some independent measure of performance to determine suitable subsets and will then employ a classification algorithm to choose the best from that subset.

It should be noted that all of the feature reduction methods discussed share the common goal of discovering a single subset of features that provides the 'best' result. The 'best' value may be measured in different ways but is frequently measured as producing the highest accuracy. The investigation suggested in this paper is somewhat different, in that the goal is to find all of the subsets which provide an acceptable answer, not just the best one. As such, the focus of this effort was somewhat different than those discussed in the filter, wrapper and hybrid feature selection papers.

The Filter algorithm

Cardie (1993) was an early proponent of using feature set reduction with classifiers. She attempted to determine the value of unknown tokens in a natural language processing. Cardie's approach was to generate a similarity metric that focuses on a suitable subset of features by identifying the attributes most important for accurate prediction. The classifier was used to determine which attributes were needed to make the decision regarding the value of the unknown token. Hers was an early example of filtering.

Kira and Rendell (1992) proposed a system called Relief as a method to reduce the size of the feature set. They asserted that it was noise-tolerant and would maintain accuracy even if the features interacted one with another. They suggested that feature set selection was important to decrease the learning time and increase the classifier accuracy. Their initial observation was that there may be many features included in a feature set that do not contribute to the result. Further, it is generally not known in advance which features contribute and which do not. Hence, more features are collected than required and the run time of the classifier will be longer than it might otherwise be. They described feature selection as "the problem of choosing a small subset of features that ideally is necessary and sufficient to describe the target concept" (p. 129).

Relief employs a threshold of relevancy as a form of threshold measure. It employs a weighting measure to determine which features' significance rises above the threshold measure and includes them in the reduced feature set (and does not include them if they are below the threshold). Kira and Rendell (1992) noted that Relief can accommodate feature interaction. The two principle components of Relief are the averaged weight vector and the threshold. They noted that the difference in relevance for features must be material (that is, high for relevant features and small for irrelevant ones). Further, the threshold level must be carefully selected. They were able

to conclude that "Relief is a useful algorithm even when feature interaction is prevalent and the data is (sic) noisy. These results show that Relief is significantly faster than exhaustive search and more accurate than heuristic search." (p. 132). Of note, Relief would not eliminate redundant features. Further, it is sensitive to the count of the training instances and could produce erroneous results if there were not enough instances.

Almuallim and Dietterich (1991) suggested that, in many domains, a bias towards using a minimal set of features, which they referred to as a MIN-FEATURES bias, should be preferred. They proposed an algorithm which could achieve this goal. Their key concept was that "if two functions are consistent with the training examples, prefer the function that involves fewer input features" (p. 547). Consistent feature subsets are ones that do not have the same values if the instances belong to different classes. They termed their algorithm 'FOCUS' and suggested applying FOCUS as a preparatory step and so removing any irrelevant features before the training data was used with a classifier. They suggested as example using FOCUS as a preprocessing step with ID3.

Almuallim and Dietterich (1991) thought that FOCUS would generalize better in some domains and require fewer training examples. Further, they asserted it would be better where many-featured training sets are being used to train multiple classifiers as there may be much irrelevant or redundant information.

Several other versions of FOCUS were developed. Almuallim and Dietterich (1992) produced FOCUS-2 which was shown to be able to handle larger feature sets and be materially faster than FOCUS and demonstrated that the performance of ID3 was improved when FOCUS-2 (and another, related) algorithm were used. The main idea in FOCUS-2 was to concentrate on

attribute subsets which were more likely to be consistent. They employed the concept of a conflict vector which is specified for each two instances belonging to different classes. The vector identifies the features that make the instances different.

Further research was carried out on the FOCUS system. Arauzo, Benitez and Castro (2003) developed C-FOCUS which extended FOCUS-2 to manage non-discrete (continuous) valued features. They altered the way the consistency was determined. For their model, inconsistency is said to occur when two feature sets that are very similar to each other belong to different classes. The limit of 'how similar' is a tunable parameter called the degree of similarity. As the user must make a decision as to the value of this parameter, results may vary significantly depending on the choice.

Santoro, Nicoletti, and Hruschka (2007) extended C-FOCUS to C-Focus-3 with the intent of improving performance. They implemented a heuristic which breaks the search space into two regions, each with their own method. The 'intermediate region (IRM)' is that portion that has subsets containing between 1/3 and 2/3 of the features and the rest is termed the 'remaining regions (RRM)'. They tested their approach on 10 well-known datasets. They observed that C-Focus-3 could be considered more efficient because in never examined more than 70% of the search space. Further, in situations where its performance was inferior, the difference was relatively minor. They further noted that their approach was especially effective when the number of relevant features represented more than half of the potential features.

The Wrapper algorithm

John, Kohavi and Pfleger (1994) examined feature set reduction with respect to both filters and wrappers. They explored the concepts of relevance and irrelevance for features and discussed the Filter Model and the Wrapper Model.

They provided four potential definitions of relevance and then proceeded to demonstrate that there were problems with each of them. They propose a two tiered notion of relevance and one of irrelevance. Strong relevance occurs when a feature's removal always results in a loss of accuracy for the classifier. Weak relevance occurs when the feature sometimes contributes to accuracy, but will not always do so. Irrelevant features do not contribute to accuracy.

John et al. (1994) reviewed several flavors of the Filter Model, including the works by Kira and Rendell (1992), by Almuallim and Dietterich (1991), and by Cardie (1993) and highlighted deficiencies with them. With respect to FOCUS, it searches for the minimum set of features to determine the correct answer. If the features happen to include a unique identifier (e.g. SSN) that feature alone would be enough to correctly classify the instance. It would not generalize well. With respect to the Relief process, John et al. (1994) remarked that it "does not attempt to determine *useful* subsets of the weakly relevant features" (p. 124). They noted that it is frequently the case that many features exhibit high correlation and so would be weakly relevant; Relief would not eliminate them. With respect to Cardie's (1993) approach there was a general concern that a bias could readily be introduced because it "ignores the effects of the selected feature subset on the performance" (p. 124) of the classifier.

John et al. (1994) concluded that the subset selection must consider the biases of the classifier in order to produce acceptable results on unseen data values. They suggested that the Wrapper Model might produce superior results.

Their Wrapper Model proposal is that feature selection should be a wrapper around the actual classifier. They suggested several greedy options. Backward elimination starts with a complete set of options and then selectively removes the one that contributes least to the solution, when employed with the classifier. Forward selection works in the reverse, where the feature set would initially be empty and then greedily adds features that improve the answer the most. They also noted that the algorithm could be improved by both adding and subtracting features in the same step. The algorithm runs in $O(n^2)$ time, with n the number of features, which would render it impractical for even modest sized feature sets.

John et al. (1994) concluded that subset selection could result in smaller structures which could lead to better understanding of the domain. In general, they found that accuracy did not improve significantly (with certain exceptions).

Kohavi and John (1997) extended and amplified the work of John et al. (1994). They noted that the problem they were addressing was that of selecting the appropriate subset of features and ignoring the remainder while recognizing that the method of selecting the subset and the classifier itself interact. Specifically, they used the wrapper method and extensively compared it to the filter method. They adopted as their goal to maximize classification accuracy on an unseen test set. They noted they might have made other choices for the goal, such as identifying and using only the relevant features.

Kohavi and John (1997) noted that many of the common classifiers degrade in prediction accuracy if many features are included in the feature set that are not actually required. That is, adding extra features is not a neutral event in terms of accuracy. Doing so is always negative in terms of execution time. However, they noted that "the optimal Bayes rule is monotonic, i.e., adding features cannot decrease the accuracy" (p 276). Further, they noted that an optimal subset may not be unique (for example, if two features are perfectly correlated one can be replaced with the other).

They examined several definitions of relevance and irrelevance and demonstrated that each of these definitions presented problems. They reiterated the recommendation of John et al. (1994). Specifically, strong relevance occurs when a feature's removal always results in a loss of accuracy for the classifier; weak relevance occurs when the feature sometimes contributes to accuracy, but will not always do so; and irrelevant features do not contribute to accuracy. Kohavi and John (1997) also demonstrated that relevance does not imply optimality and its reverse. They reiterated the criticisms of John et al. (1994) with respect to the filter approach.

Kohavi and John (1997) expanded upon the wrapper approach of John et al. (1994) and explained their (Kohavi and John's) methodology in great detail. They noted that the feature subset selection is done without any knowledge of the classifier proper (other than the interface). That is, the classifier guides the selection of features. They noted that a search space requires an initial state, a termination condition, and a search engine. They compared hill-climbing and best-first search as the search engines. They used a five-fold cross-validation evaluation, repeated multiple times, with the repetition count being guided by the standard deviation observed. The datasets were from the UC Irvine repository and included both observed and synthesized ones. They used decision-tree (ID3) and Naïve-Bayes as the classifiers.

Kohavi and John (1997) reported results for the greedy hill-climbing search with both ID3 and Naïve-Bayes. For ID3, both showed uniformly better performance with the real (naturally occurring) datasets and very mixed results with the synthetic ones. They hypothesized that the issues with the synthetic datasets might be related to high-order interactions. For these, no single addition of a new feature resulted in an improvement and therefore the hill-climbing portion terminated too early to be of effect. With respect to Naïve-Bayes and the real datasets, there was no significant difference in accuracy but that accuracy was obtained by using very few features. Naïve-Bayes did not show any advantage when used with the synthetic datasets.

Kohavi and John (1997) also used the best-first search (BFS) engine which they described as being more robust than hill-climbing. The essential feature of BFS is to expand the most promising (unexpanded) node that has been seen to date. The process stops when a stale search is observed (i.e. no improvement above some threshold for some count of expansions). Both values are tunable. They observed little difference for both ID3 and Naïve-Bayes when used with hill-climbing or BFS on the real datasets. For the synthetic datasets, only one showed material improvement with BFS.

Kohavi and John (1997) discussed the use of compound operators. Their key idea was "a new way to change the search space topology by creating dynamic operators that directly connect a node to nodes considered promising given the evaluation of its children" (p. 297). This process would reduce the number of subsets that must be evaluated. They reused their notion of strongly relevant, weakly relevant and irrelevant features and remarked that an optimal feature subset will, in reality, be composed of only relevant features (even if it is possible to contrive situations where that is not the case). Hence, the compound operator can be determined from the accuracy of the children of a particular node. The operators are ranked by the estimated accuracy of their children.

Then the compound operator c_i is the combination of the best i + 1 operators. Kohavi and John (1997) asserted that the principle advantage of using compound operators is to make backward feature selection (where features are removed rather than added) computationally feasible.

Kohavi and John (1997) also provided a discussion regarding overfitting. They noted that "Overuse of the accuracy estimates in feature subset selection may cause overfitting in the feature-subset space." (p. 311). As the number of subsets is so large, it is probable that one of them will be seen to be acceptable on the hold-out sets but yet perform poorly on new data. They provided the example of the 'no-information' dataset where the values are all generated randomly. Nevertheless, subsets can be found that produce high accuracy while training but have limited (if any) predictive power. They further asserted that, while acknowledging the problem exists, experimentally they have only observed the issue when the number of instances was small.

Kohavi and John (1997) examined the issue of subset selection as search with probabilistic estimates. They noted that it is possible to decrease the variance by performing the accuracy estimates more than once and averaging the results and so shrinking the confidence interval for the mean. Performing the operation repeatedly necessarily consumes more time. Alternately, that time could be used to make a more detailed examination of the search space. The two uses of the time are in tension. They formalized the goal statement to optimize this search and commented on several different earlier approaches. They briefly summarized five approaches by different researchers. In summary, Kohavi and John (1997) concluded that "By using search algorithms that take advantage of the probabilistic nature of accuracy estimates, it is possible to explore a larger portion of the space if the evaluation time for a state can be reduced based on statistical estimates." (p. 314). More investigation was recommended.

Kohavi and John (1997) also provided extensive comments on related work. They noted that they have observed work being done by other researchers on their wrapper approach. They also noted related work where the term 'wrapper' was not used but the main idea was essentially the same.

Recommendations for future work, and implicit criticisms, of the wrapper method were added by Kohavi and John (1997). They suggested that simulated annealing might be worth investigating. They noted that they have started their search with both full and empty sets of features, but other starting points are possible. They commented that the wrapper approach is very slow and that for larger datasets using a less computationally intensive accuracy measurement might be worthwhile. They remarked that the wrapper method lends itself to easy parallelization.

Kohavi and John (1997) concluded "In supervised classification learning, the question of whether a feature in a dataset is relevant to a given prediction task is less useful than the question of whether a feature is relevant to the prediction task given a learning algorithm." (p. 74). Therefore, the impact of the optimization goal (e.g. accuracy), the biases of the different algorithms, and the nature of the training set must be considered.

The Hybrid algorithm

Das (2001) proposed a hybrid solution for feature selection. The hybrid method described uses features of both the filter and wrapper approaches. He noted the four main issues with feature selection as "the starting point of the search, the organization of the search, the evaluation of feature subsets and the criterion used to terminate the search" (p. 74) .

Das (2001) noted that a good reason for using the wrapper method is that using the estimated accuracy of the classifier directly is the best way to measure the contribution of the

features. Further, he observed that different classifiers had significantly different optimal feature sets; hence the use of the filter method, with its feature selection performed without advice from the classifier, is problematic. However, the cost of running the classifier on each subset under consideration is computationally expensive and frequently infeasible. Hence, wrappers do not scale well as the datasets increase. Further, he noted that wrappers tend to overfit when the training sets are small. By comparison, filter methods are fast and scale to large datasets. The issues with filters and wrappers motivated his work.

Das (2001) noted several features that cause difficulty for many classifiers, specifically co-predictors, disjunctive concepts and redundant and irrelevant features. He stated that "for most real-world datasets, a feature set that allows one algorithm to induce a high-accuracy concept should also allow a different algorithm to induce a high-accuracy concept, even if the feature set selected is not optimal for that algorithm." (p. 75). He further suggested that the accuracy should be 'relatively similar' even though the feature sub-sets might be selected using different methods.

Das' (2001) proposal was to initially create a filter method that would select features "that are highly predictive of a small part of the instance space" (p. 77). Although initially designed with a pre-selected feature set size, the next phase was to allow the algorithm to increase the feature set size depending on the accuracy observed. The final improvement was to use the actual learning algorithm to direct the search.

The first instance of the algorithm used boosting but only for feature selection. The algorithm is run for a set number of rounds. At each round, the feature is selected which provides the highest information gain and that feature goes into the set to be returned. Once a feature has been selected it is not considered in any future round, that is, only unselected features are

considered at each round. Das (2001) called this algorithm Boosted Decision Stump Feature Selection (BDSFS). He noted that this method performs well, even though it is not theoretically optimal.

One issue with BDSFS is that the number of features to be selected must be specified in advance. He modified the feature selection algorithm so that the search stops when the training accuracy fails to improve. He reported that this variant, called BDSFS-2, performed well on the datasets under consideration. He also modified the way the feature selection algorithm was used. The reweighting process was altered to avoid the inefficiency of wrappers. This last method was called Boosting Based Hybrid Feature Selection (BBHFS). He reported that the algorithm was very fast and performed well on the test data.

Das (2001) tested his methods. He used k-Nearest Neighbors, Naïve-Bayes and ID3 and noted that these three have fundamentally very different methods of operation, hence form a reasonable cross-section of available algorithms. He compared the filter, wrapper and hybrid methods. The experimentation demonstrated that there was little difference with respect to the accuracies generated by k-NN, Naïve-Bayes and ID3 when the feature sets were selected by Naïve-Bayes and ID3. Hence, if one method performs well on a given feature set then the others will likely also perform well on that feature set, even if the feature set is not optimal for the other methods.

With respect to BDSFS, he found that its results were comparable to that of the wrapper approach. He found that "the performance of BDSFS-2 and BBHFS is equivalent and the accuracies they obtain are comparable to the accuracies obtained with simple BDSFS" (p. 78), with one exception. He also noted that "BDSFS-2 and BBHFS significantly outperform BDSFS

and forward selection wrappers. This is because the stopping criterion used makes the algorithm select many more features that are useful in discriminating among examples in small portion of the instance space" (p. 78). Das (2001) remarked that the stopping criteria were effective but it was unclear whether or not using the learning algorithm in the reweighting for boosting was. He noted that the speedup with BBHFS was very significant. While BDSFS-2 and BBHFS were competitive with wrappers on large training sets, he found that BDSFS-2 and BBHFS performed inconsistently on small ones; better than wrappers for some sets but worse for others.

Summary of Feature Selection: A Data Perspective

Li et al. (2017) produced a comprehensive survey of the current state of the practice of feature selection (Feature Selection: A Data Perspective). This section of the literature review will largely follow their structure and original source content, although the material will be materially abridged due to space considerations. Of note, much of their effort was with respect to areas that are not of interest to this investigation. As such, these areas will be noted and, where appropriate, very briefly summarized. However, no attempt will be made to elaborate on those sections. While interesting, they are not germane to this effort.

Li et al. (2017) noted that their interest was in selecting features such that the produced models were simpler and more comprehensible, to improve data mining performance and so that the data itself could be cleaner and more understandable. They also remarked that much of their work focused on the area of big data – either data with high dimensionality, or large count of instances, or both. Of note, this research effort did not use big data – the cardinality of the feature sets was consistently modest as were the counts of instances available. In this respect, the work of Li et al. (2017) and this research effort somewhat diverge. Nevertheless, there is still a significant amount of overlap and those areas will be discussed.

Li et al. (2017) reviewed supervised, unsupervised and semi-supervised approaches. As the current investigation was only concerned with supervised learning, no further discussion of unsupervised or semi-supervised approaches will be produced.

Li et al. (2017) used several categories of data. Data can be primarily divided as static or streaming. This investigation was only interested in static data as diagnostic tests are not performed in a streaming fashion (although some monitoring tests might be). Li et al. (2017) further divided static data into conventional and heterogeneous types. Heterogeneous data consist of linked data, multi-source data and multi-view data. As none of these were used in any of the test beds that were employed in this investigation, they will not be discussed further. Li et al. (2017) further divide conventional data into those with flat features and those with structured features. As this investigation was focused on medical diagnoses the data of interest does not generally have a structure. Therefore, the approaches of interest are traditional feature selection and those with structured features will not be considered further.

Li et al. (2017) discussed both feature extraction and feature selection. Feature extraction operates by projecting a set of features onto a smaller dimensional space. A side effect of feature extraction is that the inherent meaning of the original data may be either completely lost or simply be difficult to unambiguously interpret. However, feature extraction is not part of the current investigation and will only be mentioned briefly. As noted elsewhere, feature selection is the intentional selection of a subset of the existing features which results in a cleaner, smaller, more easily understood and processed set of features.

Li et al. (2017) noted that real-world data often contain features that are irrelevant, redundant or noisy. It was a fundamental assumption of this current investigation that their

assertion is generally and widely valid. This then was the goal of this investigation and their effort; that is, to eliminate the features that do not contribute (the irrelevant), do not provide unique information (the redundant) or are ambiguous (the noisy). As noted above, this review will consider conventional, flat, static data. Therefore, the discussion will be focused on traditional feature selection. They further remarked that, of the three main styles of feature reduction (filter, wrapper and hybrid), the wrapper methods are not generally used due to their high computational costs, especially when the dimensionality of the feature set is high.

Li et al. (2017) split the discussion of feature selection into five main areas: similarity-based methods, information-theoretical-based methods, sparse-learning-based methods, statistical-based methods, and other methods.

Similarity-based methods

Li et al. (2017) discussed a family of algorithms that utilize some similarity measure to reduce the feature set. When class labels are available, these are used to assist in determining the similarity of various features. Pairwise similarity of features is encoded in a similarity matrix. A utility function produces the utility of a feature subset. For a subset of size *k* the standard approach is to select the top *k* features that maximize the individual utility. There are several different approaches to prioritizing the features to be selected. He, Cai, and Niyogi (2005) suggested using the Laplacian score for feature selection. Theirs was a filter approach that can be used with supervised or unsupervised learning. The Laplacian score is used to determine the locality preserving power of each feature. For a given dataset a weighted graph is constructed with edges connecting 'nearby' points one with the other. The algorithm favors features with large variance demonstrating more representative power. They demonstrated their algorithm on

the UCI Iris dataset and on a facial recognition dataset and reported good results for both. Li et al. (2017) remarked that this approach "is a special case of utility maximization" (p. 94:8).

Zhao and Liu (2007) proposed a unified framework that is appropriate for both supervised and unsupervised learning. They demonstrated that ReliefF (a supervised algorithm) and Laplacian Score (which includes unsupervised features) are special cases of their algorithm. They noted that, to this point, supervised and unsupervised approaches have largely been considered separately. They remarked that a "unified framework will enable us to (1) jointly study supervised and unsupervised feature selection algorithms, (2) gain a deeper understanding of some existing successful algorithms, and (3) derive novel algorithms with better performance" (p. 1). They further argued that supervised and unsupervised learning can be considered together provided the objective is "to select features that are consistent with the target concept" (p. 1). The key is to be able so separate subsets of features using appropriate definitions of separability.

Zhao and Liu (2007) named their framework SPEC (from Spectrum decomposition). Feature relevance is related to a feature's ability to provide superior separation of the data (that is, the groups produced are more consistent with the target concept – hence the process applies to both supervised and unsupervised learning). Therefore, feature relevance is the key concept. Further, they were able to reduce the effect of noise by focusing on features which provided better separability. They were then able to establish a ranking hierarchy. They noted that some of this effort can be computationally expensive. In summary, their algorithm constructs a similarity set and constructs a graph representation of same, evaluates the features and then ranks the features in order of relevance. The time complexity is largely related to the cost of building the similarity matrix. They then demonstrated that Laplacian Score from He et al. (2005) and ReliefF from Zhao and Liu (2007) are special cases of SPEC.

Zhao and Liu (2007) evaluated their proposed methods on both supervised and unsupervised cases. For unsupervised environments, they found that the more features employed, the better the results. However, they reported that this trend was less material once more than 40 features were used. They further remarked that supervised feature selection performed better than unsupervised feature selection and surmised that the difference was largely due to the availability of label information. Specifically, they provided several different formulations of their algorithm with guidelines as to when to use which instance. In conclusion, they asserted that many existing feature selection algorithms are simply special cases of their generalized algorithm and suggested that further research might generate new algorithms from their work.

The Fisher Score is another similarity based feature reduction method, discussed by both He et al. (2005) and Duda, Hart, and Stork (2001) and presented by Li et al. (2017). He et al. (2005) noted that the Fisher Score can be considered a special case of the Laplacian score provided specific graph structures are valid. When valid, the Laplacian score, L_r, is related to the Fisher score, Fr, by the equation $L_r = 1/(1 + F_r)$.

Nie, Xiang, Jia, Zhang and Yan (2008) noted that traditional methods (such as Fisher score or Laplacian score) find a measure (score) for each feature in isolation and then select features based on that score. Hence, while it is apparent that the top-scoring features, when measured in isolation, are selected, there is no guarantee that the selected features are actually the best as a collection. They suggested using an iterative algorithm to find the globally optimal feature subset. They start with two weighted, undirected graphs. The first graph represents the within-class relationships (local affinity) while the second represents the between-class relationships (global affinity). If data x_i and x_j belong to the same class they are 'close to' each other, while if they belong to different classes then they are not as close. To deal with the within-class relationship,

the feature subset with the smallest sum of differences is desired. The opposite occurs when the data belongs to different classes: the largest sum of differences is desired. Therefore, the ratio of these two values is of interest. These relationships can be expressed as relationships between matrices. Nie et al. (2008) then transformed the problem into one of optimizing the relationship between the traces of the matrices produced. Through further manipulation, they demonstrated that the solution can be produced through an iterative process that quickly converges. They noted that relatively few steps are required to achieve the optimum result. They compared their approach to that of Laplacian or Fisher scores and reported generally, but not always, superior results. They did remark that "although our method theoretically guarantees to find the feature subset with the optimal subset-level score, it is not always guaranteed to obtain the optimal accuracy rate" (p. 675). A key observation was that the Laplacian and Fisher (and other) methods that simply greedily select the best features sequentially should not be expected to select the optimal subset.

Relief has been discussed above. Several enhancements to the basic Relief method have been proposed. Kononenko (1994) proposed several of these extensions. He noted that "the key idea of RELIEF is to estimate attributes according to how well their values distinguish among instances that are near each other" (p. 171). The original version of Relief searches for two specific neighbors, the nearest hit and the nearest miss and creates a weight based on the results with the concept being that good, or valuable, attributes should be able to "differentiate between instances from different classes and should have the same value for instances from the same class" (p. 172). One serious issue with the original version was that only two classes could be considered. Kononenko (1994) extended the original with several modifications. Noisy data can cause unreliable estimates with respect to the selection of nearest neighbors. Instead of simply

looking for the nearest neighbors, Relief-A searches for the k-nearest neighbors (with k being a user defined value) and then averages the values generated. Kononenko (1994) noted that while there was moderate improvement for noise free datasets with Relief-A, there was 'drastic' improvement for noisy datasets as the value of k was increased.

Kononenko (1994) proposed Relief-B to deal with the situation where the value of at least one of the two instances is unknown by modifying the term used to calculate the difference. Relief-C is as Relief-B except that it simply ignores the contribution if one of the values is unknown. Relief-D calculates the probability that two given instances have different values using a different equation depending on whether one or both instances have unknown values. He noted that Relief-D performed slightly better.

Kononenko (1994) also suggested some enhancements to deal with the situation where there are more than two classes. He was dissatisfied with the obvious solution of simply reforming the problem as a series of two-way decisions. Relief-E was designed as a generalization of Relief such that a "near miss of the given instance *I* is defined as the nearest neighbor from [a] different class" (p. 178). This was extended to Relief-F where instead of finding one near miss for one different class it finds one near miss from each different class and averages the results. He considered Relief-F to be the most significant of those developed as it can deal with missing data, noisy data and multi-class problems. (Of note, elsewhere, Relief-F has been termed RELIEFF.)

Li et al. (2017) summarized these methods:

> Similarity-based feature selection algorithms have demonstrated with excellent performance in both supervised and unsupervised learning problems. This category of methods is straightforward and simple as the computation focuses on building an affinity

matrix, and afterwards, the scores of features can be obtained. Also, these methods are independent of any learning algorithms and the selected features are suitable for many subsequent learning tasks. However, one drawback of these methods is that most of them cannot handle feature redundancy. In other words, they may repeatedly find highly correlated features during the selection phase. (p. 94:10).

Information-theoretical-based methods

Li et al. (2017) noted that information-theoretical-based methods attempt to maximize feature relevancy and minimize feature redundancy. Most of these algorithms are designed to work in a supervised fashion because the relevance of a feature is generally determined by its relationship with the appropriate class label. Further, these algorithms only work with discrete variables, although data manipulation can always be used to 'discretize' continuous ones.

A common concept in many of these algorithms is the notion of entropy. The entropy of a discrete random variable X is defined as:

$$H(X) = -\sum_{x_i \in X} P(x_i) \log_2(P(x_i))$$

where x_i is the specific value of the random variable X, and

$P(x_i)$ is the probability of x_i over all possible values of X (Li et al. 2017).

When $\log_2$ is used the entropy is measured in bits (Duda et al., 2001).

The conditional entropy of X given another discrete random variable Y is defined as:

$$H(X \mid Y) = -\sum_{y_j \in Y} P(y_j) \sum_{x_i \in X} P(x_i \mid y_j) \log(P(x_i \mid y_j))$$

with $P(y_j)$ as the prior probability of y_j, and

$P(x_i \mid y_j)$ as the conditional probability of x_i given y_j. The conditional entropy demonstrates the uncertainty of X given Y. Li et al. (2017) then describe information gain (or mutual information) between X and Y as

$$I(X;Y) = H(X) - H(X \mid Y) = \sum_{x_i \in X} \sum_{y_j \in Y} P(x_i, y_j) \log \frac{P(x_i, y_j)}{P(x_i)P(y_j)}$$

Li et al. (2017) noted that searching for the optimal set of features is NP-hard. As such, most algorithms they discussed used some form of heuristic sequential search. Otherwise, the search space is simply too large and the time required to search it would be too great.

Li et al. (2017) noted the work by Lewis (1992) related to Mutual Information Maximization (Information Gain). Lewis (1992) was interested in categorizing written text. This work has value in indexing texts for document retrieval, extracting data from texts, and assisting humans in accomplishing these tasks. In Lewis' (1992) approach, the value of a feature is determined by its correlation with the class label, hence this method is applicable to supervised learning situations. Lewis (1992) remarked that the "simplest indexing languages are formed by treating each word as a feature. However, words have properties such as synonymy and polysemy that make them a less than ideal indexing language." (p. 212). He proposed a number of methods by which features could be defined. These included 'syntactic indexing phrase' by which features are defined by the presence of two or more words in a particular syntactic relationship. Alternately, 'term clustering' was suggested as replacing groups of features with a single feature. A main concern was to determine the impact of feature set size on the effectiveness of categorization. Lewis (1992) noted that "a primary concern of ours [sic] was to examine the effect

of feature set size on text categorization effectiveness" (p. 213). Potential features were ranked on the basis of expected mutual information relative to the feature and the assignment for a given category. An arbitrary number of 'top' features were selected, with the count of 'top' features a topic of the study. The probabilistic model used for text categorization used to estimate the probability that category C_j is assigned to document, D_m $P(C_j=1|D_m)$, was given as:

$$P(W=1)\text{x}\prod_i \left(\frac{P(W_i=1\mid C_j=1)\times P(W_i=1\mid D_m)}{P(W_i=1)}+\frac{P(W_i=0\mid C_j=1)\times P(W_i=0\mid D_m)}{P(W_i=0)}\right)$$

with

$P(C_j=1)$ as the prior probability that category C_j is assigned to a document, in the absence of any information about the contents of the particular document.

$P(W_i=1)$ as the prior probability that feature W_i is present in a randomly selected document. The index *i* ranges over the set of predictor features for category C_j.

$P(W_i=1\mid C_j=1)$ as the probability that feature W_i is assigned to a document given that we know category C_j is assigned to that document.

$P(W_i=1\mid D_m)$ as the probability that feature W_i is assigned to document D_m.

note $P(W_i=0\mid C_j=1)$ is 1- $P(W_i=1\mid C_j=1)$

Lewis (1992) experimented with forming clusters from words using three metafeature definitions and with phrases under eight metafeature definitions. Performance was measured using both recall (the fraction of relevant instances retrieved over the total relevant instances) and precision (the fraction of relevant instances among the retrieved instances). Lewis (1992) found

that the number of features required was (to him) surprisingly low, 10 features on one set (with this set having 14704 training examples) and 15 on another (with this set having 1300 training examples). He noted that the effectiveness of the model decreased as the number of features employed increased past the values noted and suggested that it might be due to the "curse of dimensionality". Another possible explanation presented was that a key assumption might not be valid. That assumption was that "the probability of observing a word in a document is independent of the probability of observing any other word in the document, both for documents in general and for documents know to belong to particular categories" (p. 215). As the feature set increases so does the opportunity for similar terms to be identified in the documents.

Lewis (1992) was disappointed that term clustering did not significantly improve the results for either words or phrases. He noted that "many of the relationships captured in the clusters appear to be accidental rather than the systematic semantic relationships hoped for" (p. 216). He also suggested that the results might be the consequence of too little training data. Further, he suggested that his use of metadata might need to be re-evaluated in that it was too coarse-grained. Hence, a significant number of 'accidental' co-occurrences were possible. Last, he noted that 'while phrases are less ambiguous than words, they are not all good content indicators" (p. 217). Li et al. (2017) noted that in Mutual Information Maximization, the features are assessed independently one of another. Hence, feature redundancy is ignored.

Mutual Information Feature Selection

Li et al. (2017) referenced the work by Battiti (1994) in which he described the use of the mutual information criterion as a method for the evaluation of candidate features and to guide the selection of a subset of those features. One key feature is that this method considers the mutual information between input variables. Therefore, when selecting the next feature to add to the

already-selected ones that feature should be strongly correlated with the class labels but weakly correlated with the already-selected ones. That is, if the next feature to be added should be the one that contributes the most new information to the selection process. He suggested a greedy algorithm that would generate a feature subset by taking advantage of the mutual information of the not-yet-selected features and the already-selected ones.

Battiti (1994) noted that his method is a pre-processing one and results in a selection of *k* (*k* is user defined) features. His main objective was to reduce the dimensionality of the initial feature set to reduce the complexity of the classifier and improve the classifier performance. He noted that "Feature selection methods that are sufficient for simple distributions of the patterns belonging to different classes can fail in classification tasks with complex decision boundaries." (p. 537). Battiti (1994) referenced Shannon's Information Theory which specified the entropy and conditional entropy as noted by Li et al. (2017), above.

Battiti (1994) noted that the mutual information is the amount by which uncertainty is decreased and is (by definition):

I(C:F)=H(C) - H(C|F), with the relationship being symmetric.

Using

$$I(C;F) = I(F;C) = \sum_{c,f} P(c,\boldsymbol{f}) \log \frac{P(c,\boldsymbol{f})}{P(c),P(\boldsymbol{f})}$$

with $\boldsymbol{f}$ the input vector and c the class.

Battiti (1994) then framed the problem as: "Given an initial set *F* with *n* features, find the subset $S \subset F$ with *k* features that minimizes $H(C|S)$, i.e., that maximizes the mutual information

$I(C;S)$" (p. 539). He then commented that there are two serious computational issues with solving this problem; the number of examples may be impractical and the processing time required may be infeasible. He then suggested an alternate approach of which requires fewer resources. Instead of using the entire feature vector and the class, the calculations would be performed between individual features. Then the subsets would be generated using a greedy approach. This algorithm penalizes features that have information similar to that already in the subset selected to date. He called the algorithm Mutual Information Based Feature Selection (MIFS). Several test scenarios were developed to demonstrate the effectiveness of his approach. He noted that "although the availability of sufficient information does not guarantee the convergence of a neural net training algorithm to a satisfactory performance level, we presented some examples in different classification areas where the method is satisfactory" (p. 548).

Peng, Long, and Ding (2005), proposed a Minimum Redundancy, Maximum Relevance approach to selecting an optimal subset of features. They noted that the optimal characterization condition typically means the minimal classification error. They commented "In an unsupervised situation where the classifiers are not specified, minimal error usually requires the maximal statistical dependency of the target class c on the data distribution in the subspace R^m (and vice versa)" (p. 1226).

Peng et al. (2005) noted maximal relevance (Max-Relevance) is a common approach to achieving Max-Dependency, which is to select the features with the highest relevance to the target class c. They point out that good individual features do not always combine to produce a good subset of features ("the m best features are not the best m features" (p. 1226)). They noted that the minimal redundancy (Min-Redundancy) has been suggested as a method of feature selection, whereby new features are selected such that they are 'dissimilar' to the existing subset selected.

Peng et al. (2005) describe Max-Dependency as:

$maxD(S,c)$, with $D=I(\{x_i, i=1,\ldots,m\};c)$

with S the feature set with m features $\{x_i\}$ which jointly have the largest dependency on the target class c. Two serious problems with Max-Dependency in high-dimensional space are that there may be too few samples available and that the computations themselves may be problematic. Hence, its use may be restricted to selecting only a small number of features or in situations where high accuracy is not critical to the outcome (Peng et al. 2005).

In an attempt to avoid the issues with Max-Dependency Peng et al. (2005) suggested using maximal relevance (Max-Relevance) as the criterion. It was defined as:

$$maxD(S,c), \text{ with } D = \frac{1}{|S|}\sum_{x_i,x_j \in S} I(x_i,x_j)$$

Peng et al. (2005) suggested that the redundancy among features so selected could be very high and felt that little additional class-discriminative power would be provided with redundant features. Hence, if some of these were removed, the class-discriminative power would not be materially altered. Hence, they suggested the minimal-redundancy as:

$$\min R(S), \text{ with } R = \frac{1}{|S|^2}\sum_{x_i,x_j \in S} I(x_i,x_j)$$

Peng et al. (2005) combined the two constraints into the mRMR as the following:

$$\max \Phi(D,R), \text{ with } \Phi = D - R$$

Given that a partial subset has been selected, the process simply involves selecting the next feature, from the subset of not-yet-selected features, such that it maximizes $\Phi(.)$. Peng et al. (2005) described this type of selection as 'first-order', implying one feature being added at a time. They proved that, for this first-order incremental search, mRMR is equivalent to Max-Dependency. A remaining issue is how to determine the optimal number of features to include. They proposed a two-stage feature selection algorithm where, in the initial stage, the candidate feature set is generated using mRMR then, in the second stage, an alternate routine is used to select a compact subset of the initially chosen features. Selecting the initial subset involves determining a large number of candidate features. This is a filter step. The second stage is a wrapper step, used to reduce the number of features.

Peng et al. (2005) tested their proposed methods using Naïve Bayes (NB), Support Vector Machines (SVM) and Linear Discriminant Analysis (LDA) using multiple datasets. They concluded that "mRMR is computationally much more efficient than MaxDep" (p. 1232). Further, mRMR tended to outperform MaxDep, especially when the number of features was large. They suggested the reason was that "in high-dimensional space, the estimation of mutual information becomes much less reliable than in two-dimensional space, especially when the number of data samples is comparatively close to the number of joint states of features" (p. 1233). As mRMR is less complex and more accurate than MaxDep, Peng et al. (2005) recommended using it. They noted comparable results compared to MaxRel and came to the same conclusion. Further, they noted that "a well-designed filter method, such as mRMR, can be used to enhance the wrapper feature selection, in achieving both high accuracy and fast speed" (p. 1236). Further, they commented that their algorithm could be particularly useful where the number of features was very large (that being in the range of thousands of features). They also noted that all of their

methods selected were heuristic. None could guarantee the global maximization of a criterion function. This issue is essentially related to the size of the search space. Hence, while "mRMR seems to be a practical way to achieve superior classification accuracy in relatively low computational complexity" (p. 1236) it does not guarantee an ideal solution.

Lin and Tang, (2006) proposed an alternate method for feature selection. Their work was focused on high-dimensionality problems. They noted that many existing approaches assumed that distributions (of feature values) were assumed to be Gaussian but that the opposite might be the case. They commented that their model "points out that *the redundancy can be factorized into class-relevant and irrelevant ingredients* and introduces the concept *class-relevant redundancy* with theoretically well-founded formulation" (p. 69) (italics theirs). They also noted that their methods can drastically reduce the computational cost from $O(n^2)$ to $O(n)$.

Lin and Tang, (2006) suggested using a projection (kernelization) to extract nonlinear features such that each feature could be could be considered a projection of that mapping. They commented that information stems from uncertainty, with the mutual information $I(x;y)$ as:

$I(x:y) = H(x) - H(x|y)$, indicating that the information delivered from x to y equals the reduction of uncertainty of y when x is known. They then described the 'infomax principle' as suggesting "to learn features by maximizing the mutual information between the features and the classes" (p. 70). They established that "the information solution is near optimal in the minimum Bayes error sense" (p. 71).

Lin and Tang, (2006) noted that relations may exist between features, examined the two-feature case and then expanded that to the more general case. They remarked that "when two features are used, the joint information of the feature set would be less than the sum of information

conveyed by individual features due to the redundancy, which is measured by the mutual information between the two features" (p. 71). This equation is given as:

$H(y^{(1)}y^{(2)}) = H(y^{(1)}) + H(y^{(2)}) - I(y^{(1)}; y^{(2)})$, indicating there is a level of redundancy between the two features. Hence the "joint class–relevant information equals the sum of the individual feature information minus the total pairwise redundancies" (p. 72). Generalizing to a larger feature set, Lin and Tang, (2006) noted that the features can be extracted sequentially using what they termed as the Conditional Informative Objective. To extract the *t*-th feature, the expression

$$\Theta_t = \arg\max_{\Theta_t} \left\{ I(y^{(t)}; c) - \sum_{u=1}^{t-1} R_c(y^{(u)}; y^{(t)} \right\}$$

is optimized, with Θ the parameter for the *t*-th feature. They termed this extraction the Conditional Informative Feature Extraction. They asserted that their approach provided additional insight into the selection problem, introduced the novel concept of class-relevant redundancy and achieved "a good trade-off between the accuracy and the complexity" (p. 73).

Lin and Tang, (2006) then developed a formula to determine class-relevant information and remarked that there were two types of interactions: those between samples in the same class and those between any pair of samples and suggested an optimization that could improve the discriminatory power of their approach. They went on to divide the search space into specific regions and then demonstrated that their objective function could be greatly simplified by considering only the terms within a small local region; they called this the Local Active Region. A computational simplification reduced the time complexity from $O(n^2)$ to $O(n)$. They concluded that their approach was robust, accurate and reduced the risk of overfitting.

Lin and Tang, (2006) further proposed a method of combining the selected features to produce a final decision. They commented that one common method is to "directly compute the Euclidean distance in the feature space, and classify a sample to the nearest class" (p. 76). They noted that this approach may fail to optimally utilize the features. Instead, they employed a multivariate logistic regression model to generate a conditional probability for the classification and asserted that it provided a good balance between sparsity and the discriminative process. They then combined their processes into a unified solution.

Lin and Tang, (2006) tested their approach using a trivial synthetic problem and a face recognition problem. The trivial problem was largely for concept demonstration. The face recognition solution was compared to other commonly used algorithms. Their solution materially outperformed the other approaches as measured by the error rates. No comment was provided regarding run times.

Meyer, Schretter, and Bontempi, (2008) proposed a filter approach for feature selection suitable for environments with a large number of variables and relatively few samples. They named the selection criterion Double Input Symmetrical Relevance (DISR) and the implementation itself Matrix of Average Sub-Subset Information for Variable Elimination (MASSIVE). DISR combines two elements. Firstly, a combination of features may return more information than the sum of that from each individual feature (termed variable complementarity). Secondly, unless there is a reason to believe otherwise, they argued that it is reasonable to "assume a combination of the best performing subsets of d-1 variables as the most promising set" (p. 3). DISR can be reduced to the Densest Subgraph Problem (aka Dispersion Sum Problem). This observation led Meyer et al. (2008) to design MASSIVE using a Backward Elimination combined with Sequential Replacement (BESR) strategy. Using both observed and synthesized data they

found that their technique was competitive with other existing methods (better under some circumstances, worse under others).

Meyer et al. (2008) noted that the complementarity problem is easily demonstrated with the XOR problem ($X_1 \oplus X_2$). Knowledge of either X_1 or X_2 provides no knowledge of the outcome. However, knowledge of both provides complete knowledge of the outcome and so demonstrates complementarity.

Fleuret (2004) proposed a method for feature selection based on conditional mutual information. His was a filter approach. He noted that most filter approaches rank features according to some metric and commented that "such a ranking does not ensure weak dependency among features, and can lead to redundant and thus less informative selected families" (p. 1531). He designed his approach to iteratively select features such that the mutual information was maximized, conditionally to the response of any feature previously selected. Hence, a new feature that is 'similar' to existing features will not be selected, even if, in isolation, it appears to yield strong predictive power. He termed this the Conditional Mutual Information Maximization (CMIM) criterion. He asserted that this method provided a good trade-off of independence and discrimination.

Fleuret's (2004) tests were exclusively binary (the results were 0 or 1). His implementation was designed for feature sets with a very large number of features (40,000, in 1 example) of which only a very small number were used (50, in the same example). As with all of the methods described, his was looking for a single subset of features that produced a 'good' result. He described "The main goal of feature selection is to select a small subset of features that carries as much information as possible" (p. 1533). He also noted that to find the (absolute) best

result was infeasible for large feature sets (feature sets having ~2^n combinations to consider). He stated that, at the opposite extreme, random sampling might achieve independence between the features, especially if representation of features was represented in a balanced manner. While he noted some issues that he thought could be resolved, he asserted that the "main weakness of this approach is that although it takes care of individual performance, it does not avoid at all redundancy among the selected features" (p. 1534). Hence, he proposed his 'intermediate' solution. That is, a feature-under-consideration is good if and only if it carries significant information about the classification and that information has not yet been captured by any member of the 'subset-so-far'.

Fleuret (2004) observed that it was trivial to select a task for which CMIM would fail (e.g. a situation where the positive population was a mixture of two sub-populations with some of the features providing information about one sub-population while the rest provided information about the other. If there was a statistically dominant sub-population, features from the other sub-population might never be selected.). In the tests he performed, he found that CMIM was the best feature selection method for all cases except one.

Vidal-Naquet and Ullman (2003) suggested a method for feature selection in a binary classification within an object recognition environment which they referred to as 'informative fragments'. They noted that "when the image intensity values are used as the basic features, the separating surface between class and non-class images is usually highly non-linear and therefore difficult to learn or to approximate" (p. 1). They commented that another approach has been to develop more complex classification methods that do not require linear separation between the classes. They suggested that a tradeoff between complexity of features and the classification scheme might be informative. They noted that if the features are simple then a large number

might be required. Alternately, if features were more complex, then fewer would be needed. Hence, they suggested that using 'fragments' of the space might be useful, with a fragment being a collection of local values. Such a collection would be a localized and complex organization of the underlying data. They compared two feature types and two classification schemes and discovered that "simple features require a more complex classification function that relies on higher order aspects of the features distribution" (p. 2). If the features contain more information, then the learning process is simpler and a simple linear separator may be sufficient for optimal classification.

Vidal-Naquet and Ullman (2003) described the essentials of their process as: "Generate a large set of candidate fragments $\{F_i\}$, Compute, for each fragment, the optimal threshold that determines the minimum visual similarity for it to be detected in an image…, Select a set of maximally informative features" (p. 2). Essentially, the idea is to find informative subsets of the individual features that, taken together, provide a good solution. The fragments are determined by use of a similarity measure and a detection threshold, with a 'sliding window' over the image used to detect the presence (or absence) of the feature. Fragments are ranked with respect to their information. The fragments are then added in the order that produces the highest increase of information.

Vidal-Naquet and Ullman (2003) compared the fragment approach to that of wavelet transform. A wavelet transform can be used for object recognition. Wavelet functions have been used for pedestrian, facial and automobile detection. This transform uses frequency and orientation characteristics within an analysis window of differing scales and a kernel function selected by the sensitivity of the function to the sought after visual features.

Vidal-Naquet and Ullman (2003) tested the performance of fragments and wavelets and found that fragments performed materially better within their testing environment. They suggested that their approach of extracting a set of information rich features, selected with consideration of the class to be recognized, was supported by their research. Further their research indicated that

> "for simple generic features the classifier had to use higher-order properties of their distribution. Conversely, when the individual features were by themselves informative, the relative contribution of the higher-order interactions was reduced and a linear decision rule was enough for efficient classification." (p. 288)

Hence, the use of collections of features, as suggested by Vidal-Naquet and Ullman (2003), can be seen to be of value.

Jakulin (2005) suggested interactions between features can be informative. He described a two-way interaction as that when two attributes contribute more when taken together than they do individually. Similarly, a k-way interaction occurs when more information is obtained using the k features together than when some l ($l<k$) features are taken individually. He suggested that the concepts of mutual information, information gain, correlation, attribute importance, association and others are simply special cases of the interaction concept.

Jakulin (2005) noted that the number of potential combinations would present an intractable problem if k was large. He further commented that this value could be constrained to be relatively small which would improve performance and reliability without having material negative impact.

Meyer and Bontempi (2006) proposed a novel filter method for feature selection. Their method was designed for classification efforts where there were a very large number of features and frequently a small number of samples. They noted that their objective was to select the (single) best subset of variables that will produce the best predictive model in a supervised environment. They termed their criterion 'double input symmetrical relevance' (DISR). They noted a "combination of variables can return more information on the output class than the sum of the information returned by each of the variables taken individually" (p. 92). They described this as 'variable complementarity'.

As stated, Meyer and Bontempi's (2006) goal was to select the most relevant features while at the same time avoiding variables that were redundant. They provided several examples that demonstrated that it is difficult "to predict, in terms of relevance, the joint effect of several input variables on an output variable" (p. 94). So, without further information, two variables might be redundant or complimentary. They defined the complementarity of two random variables X_i and X_j with respect to an output Y as:

$$C_y(X_i, X_j) = I(X_{i,j}; Y) - I(X_i; Y) - I(X_j; Y)$$

with $X_{i,j} = \{X_i, X_j\}$.

Then, two variables are complementary if their complementarity with respect to Y is positive. A negative value implies redundancy.

Meyer and Bontempi (2006) compared their approach to several others (variable ranking, relevance criterion, minimum redundancy maximum relevance, and conditional mutual information maximization). They specifically addressed the concept of complementarity. They

described the lower bound on the mutual information between a subset X_S and the target variable Y as the average of the same quantity computed for all the sub-subsets $X_{S-i}=X_S \backslash X_i$ of X_S. They determined that the optimum subset was that with two elements. They then generated the definition of: given two random variables X, Y, a joint probability p(x,y), the symmetrical relevance SR(X,Y) is defined as:

$$SR(X;Y) = \frac{I(X,Y)}{H(X,Y)}$$

Then

$$X_{DISR} = \arg\max_{X_i \in X_{-S}} \{ \sum_{X_j \in X_x} SR(X_{i,j};Y) \}$$

They remarked that the "main advantage of using this criterion for selecting variables is that a complementary variable of an already selected one has a much higher probability to be selected than with other criteria" (p. 100).

Meyer and Bontempi (2006) tested their method using 11 datasets against four other approaches. They found that their method outperformed on average accuracy. Further, it was in the top two best methods for seven of the 11 datasets. They concluded that this approach might be promising in high feature-to-sample ratio classification tasks.

Yu and Liu (2003) proposed a novel filter technique. They suggested it would be able to deal with both relevant and redundant features and that it would also execute very quickly. Their approach was designed for situations with a large number of features (e.g. genome projects, text categorization, image retrieval or customer relationship management). They noted that such sets typically contain a significant number of features that are irrelevant or redundant. Eliminating

such features provides a clearer understanding of the data and typically results in better overall accuracy and faster performance. Hence, they conclude that a 'good' feature set is "one that contains features highly correlated to the class, yet uncorrelated to each other" (p. 858). They further define an individual feature as being 'good' if "it is relevant to the class concept but it is not redundant to any of the other relevant features" (p. 858).

Yu and Liu (2003) used the standard definition of entropy of a variable X as:

$$H(X) = -\log \sum_i P(x_i) \log_2 (P(x_i))$$

and the entropy of X after observing values of another variable Y as:

$$H(X \mid Y) = -\sum_j P(y_j) \sum_i P(x_i \mid y_j) \log_2 (P(x_i \mid y_j))$$

Then, the *information gain* is the amount by which the entropy of X decreases and reflects the additional information provided by Y about X and is given by:

$$IG(X \mid Y) = H(X) - H(X \mid Y).$$

Therefore Y is deemed to be more correlated to feature X than to feature Y if IG(X|Y)>IG(Z|Y). Yu and Liu (2003) defined symmetrical uncertainty (SU) as:

$$SU(X,Y) = 2[\frac{IG(X \mid Y)}{H(X) + H(Y)}]$$

which has values in the [0,1] range, with 0 indicating independence and 1 indicating that X is determined by Y (and the opposite). Then SU can be used as the 'goodness' measure. Yu and Liu (2003) remarked that this requires two concepts: relevancy and redundancy. Relevancy can

be determined with the use of a threshold level (which is admittedly arbitrary but is commonly used). Redundancy poses a more challenging problem and involves pairwise correlations between all features, typically called F-correlations and has $O(N^2)$ complexity. The question then becomes, given two features, is the level of correlation sufficient so that one feature may be removed without excessively degrading the solution. As the F-correlations are captured by the SU values it is also necessary to establish a threshold for these values. The value of $SU_{i,c}$ establishes the extent to which F_i is predictive of the class C. Similarly, it is possible to determine the extent to which F_i is correlated to each of the remaining relevant features. They defined the concept of predominant correlation as given a feature F_i and a class C the feature F_i is predominant *iff* $SU_{i,c} > \delta$ (for some value δ) and $\forall F_j \in S'(j \neq i)$ there is no F_j such that $SU_{j,i} >= SU_{i,c}$. Further, a feature is predominant to its class *iff* its correlation to its class is either predominant or may be made so by removing redundant peers. Hence, a given feature is 'good' if it is predominant in predicting the class concept. Further, Yu and Liu (2003) remarked that "feature selection is a process that identifies all predominant features to the class concept and removes the rest" (p. 859).

Yu and Liu (2003) then labelled their approach as Fast Correlation-Based Filter (FCBF). Their algorithm finds a set of predominant features (S_{best}) by finding the relevant features, ordering them, and then removing the redundant features. Ten datasets were used to test their approach and compared to four common feature selection algorithms. They found that their approach, FCBF, was materially faster the other approaches and required the use of fewer features to obtain results which were marginally superior or comparable to that of the other methods.

Li et al. (2017) summarized these methods:

> Unlike similarity-based feature selection algorithms that fail to tackle feature redundancy, most aforementioned information-theoretical-based feature selection algorithms can be unified in a probabilistic framework that considers both "feature relevance" and "feature redundancy." Meanwhile, similar as similarity-based methods, this category of methods is independent of any learning algorithms and hence are generalizable. However, most of the existing information-theoretical-based feature selection methods can only work in a supervised scenario. Without the guide of class labels, it is still not clear how to assess the importance of features. In addition, these methods can only handle discrete data and continuous numerical variables require discretization preprocessing beforehand. (p. 94:14).

Sparse-Learning-Based Methods

Li et al. (2017) also discussed sparse-learning-based methods. They noted that, while this is an area of current interest, it directly uses the classifier in order to select the feature subset. Further, related to the complex calculations required, the computational cost is frequently prohibitive. Therefore, these methods will not be discussed further.

Statistical-Based Methods

Li et al. (2017) commented on the use of statistical measures in feature selection. They remarked that the use of such measures was often employed as a pre-processing step. As such, they are not germane to the investigation at hand. However, as these terms are so commonly used, and occur elsewhere in these discussions, they will be very briefly addressed. The following is a paraphrase of their presentation.

Li et al. (2017) discussed low variance. Features below a (user-defined) threshold are eliminated. As an example, they noted that if the variance of a feature is zero, then it does not contribute any discriminatory power (a variance of zero implies that all of the values are identical). Hence, that feature should be removed. Given Boolean features, they calculated the variance score as

$$variance\_score(f_i) = p(1\text{-}p)$$

with p being the percentage of instance where the feature is 1 (again noting that this is valid for Boolean features). The features with *variance_score* falling below some predetermined value are then eliminated.

Li et al. (2017) briefly mentioned the t-score. They noted that it is used for binary classification problems.

The equation is given as:

$$t\_score = |\mu_1 - \mu_2| / (\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2})^{\frac{1}{2}}$$

with

μ_1 and μ_2 the mean values for the two classes,

σ_1 and σ_2 the respective standard deviations,

and n_1 and n_2 the respective number of instances from each class.

The t-score is a measure of how different the means of the two classes actually are. The higher the t-score, the more significant the difference.

Li et al. (2017) also briefly introduced the concept of the Chi-Square score. This test attempts to determine whether or not a given feature and the class label are independent. The Chi-Square score is given as:

$$Chi\_square\_score(f_i) = \sum_{j=1}^{r}\sum_{s=1}^{c}\frac{(n_{js} - \mu_{js})^2}{\mu_{js}}$$

with f_i a particular feature with r different feature values

n_{js} the number of instances with the jth feature value for feature f_i

$$\mu_{js} = \frac{n_{*s}n_{*j}}{n}$$

with n_{j*} being the number of instances for the jth feature with feature f_i

and n_{*s} being the number of instances in class r.

A higher the Chi-square score suggests that the feature is more important.

Li et al. (2017) mentioned the Gini Index. They commented that the Gini Index was a common term to quantify the ability of a feature to separate instances from different classes. The Gini Index is given as:

$$gini\_index\_score(f_i) = \min_{W}\left(p(W)(1 - \sum_{s=1}^{c} p(C_s \mid W)^2) + p(\overline{W})(1 - \sum_{s=1}^{c} p(C_s \mid \overline{W})^2) \right)$$

with f_i a feature with r distinct values,

p(.) denoting probability,

W the set of instances where the feature value is less than or equal to the jth feature value,

and $\overline{W}$ the set of instances where the feature value is greater than the *j*th feature.

Hence, the *j*th feature can be used to create two disjoint subsets (<= and > the *j*th value). Li et al. (2017) noted that, when used for binary classification, the maximum value would be 0.5 and that the lower the value, the more relevant the feature. They also commented that the Gini Index is not restricted to binary situations.

Li et al. (2017) also briefly discussed Correlation-based Feature Selection (CFS). The CFS score is calculated as:

$$CFS\_score(S) = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

with S the subset with *k* features,

$\overline{r_{cf}}$ the mean feature-class correlation

and $\overline{r_{ff}}$ the average feature-feature correlation.

The value $k\overline{r_{cf}}$ represents the predictive power of the feature set. $\sqrt{k + k(k-1)\overline{r_{ff}}}$ represents the amount of redundancy in the feature set. Li et al. (2017) commented that "The basic idea is that a good feature subset should have strong correlation with class labels and are weakly intercorrelated." (p. 94:19). They then noted that such calculations are computationally prohibitive so suggested a heuristic whereby a best search strategy is employed, features are added based on that with the highest utility and continuing until a stopping criteria is satisfied.

Other Methods

Li et al. (2017) very briefly summarized what they referred to as 'other methods', which included hybrid feature selection as well as deep-learning-based and reconstruction-based methods. Hybrid feature selection has been discussed above so will not be discussed further.

Li et al. (2017) noted that deep learning is not typically used for feature selection but has been so used on occasion. In particular they referenced the work by Li, Chen and Wasserman (2015). Li et al. (2015) proposed a deep learning model that would, amongst other things, perform feature selection. They suggested a "sparse one-to-one layer, where each input feature is weighted, is added between the input and the first layer" (p. 207). One of the proposed advantages of this method was that features could be selected at the input level and hence features with non-linear behaviors could be identified. They termed their model Deep Feature Selection (DFS). They noted that as the input would be sparse; only the features with non-zero weights would be selected. They stated that any sparse regularization term could be used but they selected elastic-net (a variant of LASSO), with the regularization given as:

$$\lambda_1\left(\frac{1-\lambda_2}{2}\|w\|_2^2+\lambda_2\|w\|_1\right)$$

They observed that a single subset of features would always be returned for multi-class problems (given fixed parameter settings) and that the model would automatically identify non-linear features, given its deep structure.

Li et al. (2015) tested their model on a dataset with 93 features, 3 classes and 2156 samples per class. They tested a 'deep' DFS model with a structure consisting of $\{93 \to 93 \to 128 \to 64 \to 3\}$. They compared it to a 'shallow' DFS model with structure of $\{93 \to 93 \to 3\}$, to an elastic-net (a variant of LASSO) with a structure of $\{93 \to 3\}$, and to a random forest. They determined that the best accuracy was obtained with the random forest,

followed by the deep DFS, then the shallow DFS, and the poorest was with the elastic-net. They also noted that the deep DFS took an order of magnitude more time to execute than the other three.

Li et al. (2015) noted that the random forest classifier returns the importance of each feature selected and therefore was chosen as the benchmark. They observed that most of the key features selected by their DFS models (shallow and deep) were ranked highly by random forest. Further, both the shallow and deep DFS models were much sparser than the elastic-net; they considered this sparseness a significant advantage. For a given number of features selected, the accuracy level for the DFS models was much higher than that for the elastic-net.

Li et al. (2017) briefly summarized the data reconstruction error approach. Their definition was "It defines feature relevance as the capability of features to approximate the original data via a reconstruction function." (p. 94:21). They referenced the work by Masaeli, Yan, Cui, Fung, and Dy (2010). Masaeli et al. (2010) noted that while the feature transformation approach of Principal Feature Selection is useful for dimensionality reduction, it does not provide insight into which features are significant. They proposed a method they called Convex Principal Feature Selection (CPFS). They relaxed and reformulated the problem as "a continuous optimization problem that minimizes a mean-squared-reconstruction error…and considers feature redundancy into account" (p. 619). They did so by minimizing the reconstruction error. They described their approach to resolving feature redundancy as:

> We translate PCA into a feature selection formulation as follows. If the selected subset F_{sel} is
>
> $$F_{sel} = \{f_{q1}, f_{q2}, \ldots, f_{qq}\},$$

then to reconstruct X means that: (1) every column in the reconstructed X, denoted as $\tilde{X}$, that corresponds to $f_{qj} \in F_{sel}$ should be equal to the one in X, and (2) the columns in $\tilde{X}$ that corresponds to features not selected should be the projections of the unselected features to the subspace spanned by F_{sel}. (p. 621)

Masaeli et al. (2010) developed an optimization objective function given as:

{reconstruction error} + λ {sparsity term}

Then λ can be used to adjust the relationship between reconstruction error and sparsity. With λ=0, all of the features would be selected. With a larger λ, fewer features will be selected, with none being selected for very large values of λ. Therefore, it is a trivial matter to select the number of features desired.

Masaeli et al. (2010) evaluated their approach. By using different values of lambda and graphing the results it was trivial to determine the importance of the various features. They also compared their approach to several others using nine datasets. For all but one of these sets, CPFS had the best performance with respect to classification errors.

Summary of the Literature Review

There were eight objectives for this literature review. The use of Neural Networks as a classifier for medical diagnoses was established. Second, it has been observed that there is essentially no consideration given, in the general sense, to the cost of determining a diagnosis. Third, it was established that decision trees also have a long history of success when used for medical diagnosis. Fourth, active versus passive classifiers were reviewed. While there was some consideration of costs with respect to the active classifiers, the costs were a dynamic part of the

classification process. Such a use is completely dissimilar from the approach that will be used in this investigation. Fifth, a review of early filter approaches was discussed. Sixth, a review of wrapper approaches was outlined. Seventh, hybrid approaches were briefly discussed. Eighth and last, portions of the extensive review by Li et al. (2017) were discussed. In particular, selected papers from similarity-based methods, information-theoretical-based methods, sparse-learning-based methods, statistical-based methods and other methods were reviewed.

From this limited review, several observations emerge. Both Neural Networks and Decision Trees are suitable candidates for performing diagnoses. There has been no consideration given to costs. All of the methods examined have suggested some approach to finding a single 'good' subset of features. There have been no observed approaches that suggest using costs as a material input factor. There have been no approaches that have adopted a 'find a collection of subsets that provide a 'good enough' solution'. This suggests that the approach suggested in this investigation is novel.

Chapter 3

Methodology

Overview of Methodology

In this subsection, the overall approach to the methodology used will be summarized. The approach used for identifying acceptable models will be detailed. There were multiple criteria that might have been used. Arbitrarily, a combination of overall accuracy, total positive rate and total negative rate was used when suitable. Overall accuracy was used when TPR and TNR were not suitable. Other approaches will be mentioned. Both NNs and DTs were assessed.

The approach for resolving the Research Questions, where not otherwise mentioned, are discussed here. There were five questions; these were the major focus of this investigation.

The method for searching the potential subset space will be discussed. A breadth first search would have been infeasible because of memory constraints hence a depth first search approach was taken.

The experimental setup that was employed is described. The features were assessed to establish a ranking (least significant to most significant). Several approaches to ranking were taken, discussed below. Features can be eliminated based on most important first, least important first or random order. All three approaches were tried and the results compared.

Different configurations (e.g. the number of nodes for NNs) were evaluated. The evaluations were carried out with all features present. Once a suitable configuration was established for a given dataset and classifier, all evaluations within that dataset used the same configuration.

An approximate 'best results' was established by using all features while recognizing that there would probably be some noisy values whose elimination might improve the results. A threshold limit for acceptability was established relative to the 'best results'. For example, assume that the best results obtainable were 0.900. Then, if the relative thresholds were 0.99, 0.95 and 0.93, these would be relative to 0.900 (not to 1.000). So they would be 0.891, 0.855 and 0.837 respectively. Features were eliminated using a depth first search until the performance of the remaining subset fell below the acceptability level.

The datasets used will be noted; five were used, of which two were synthetic and three were natural. All natural datasets used remain publicly accessible. No human subjects were used.

Finally, a brief summary will be provided.

Method for Identifying Acceptable Models

There are two major components that should be considered when identifying acceptable models: what element or measure of accuracy (performance measure) is being considered and the level of accuracy that is acceptable within that measure (the accuracy threshold).

The measures of accuracy that might be considered include the false positive rate, the false negative rate, the true positive rate, the true negative rate and the overall level of accuracy or some combination of these. All of these have merit depending on the circumstances. Suppose a clinic was performing an initial screening and wanted to be sure that nobody who was positive (for some

disease) was missed. The concern in this case would be to have the false negative rate as low as possible (or, conversely, have the true positive rate be as high as possible). Suppose a drug company wanted to test a drug and hence needed individuals that suffered from a specific condition. Their concern would be that the false positive rate should be extremely low (or that the true positive be as high as possible) – missing some individuals that were positive would have little consequence for such a study provided enough clinically positive patients were obtained for the study. Other measures are approached in a similar fashion. Any measure could be an appropriate criteria, depending on circumstance. It is not the purpose of this investigation to evaluate the different criteria, rather to select one and proceed on that basis.

Let ACC^0 be the overall accuracy obtained using all input data for a given classifier. Then TPR^0 and TNR^0 are similarly defined for Total Positive Rate and Total Negative Rate. Then, a model is deemed qualified for some threshold t ($0<t<1$) if $ACC>t*ACC^0$ AND $TPR > t*TPR^0$ AND $TNR > t*TNR^0$. Admittedly, this is, and must be, an arbitrary choice. Where TPR and TNR were not appropriate, only ACC was considered.

The next item considered was the level of accuracy that will be considered acceptable, that is, the value of 't' in the above paragraph. Again, this was an arbitrary choice simply because there is no absolute value to which one might appeal. It was assumed that the accuracy level obtained using all features would be near to the maximum obtainable using an ideal subset of features (recognizing that there is no agreed upon method to determine what such an 'ideal' subset might be). As the threshold value was lowered, the number of subsets evaluated increased dramatically. Further, it is unlikely that any user would be interested in a subset that provided a very poor result. That is, if (say) the ACC^0 is 90% overall accuracy, 80% might be of interest in some circumstances, it is possible to imagine that 70% might be useful in some extraordinary

situation, but it would be difficult to contrive a situation where 50% accuracy would be of interest, regardless of the cost of obtaining same. Therefore, there is a practical lower limit, even if one cannot easily quantify it. It also should be remembered that this exercise is a proof-of-concept, not a production grade investigation. The threshold value, 't', will necessarily be in some reference to the values observed when all features are present (ACC^0, TPR^0, and, TNR^0) . If the threshold value was very close to 1 then a relatively small number of subsets needed to be evaluated. However, there may be little advantage in having a set of acceptable subsets where such a set is very small. Eliminating one or two features might, or might not, offer a material cost advantage. This suggested a broader approach, that is, a lower threshold value. The initial, arbitrary, estimates for t were 0.99, 0.96 and 0.93. Some experimentation was required to establish values so that acceptable run times were observed (that is, the number of evaluations was not excessive) and useful information could still be obtained. Therefore, the final values employed for t varied materially from those initially proposed and depended on the dataset and the number of evaluations performed. In general, the values of t were increased.

Neural networks were used for the initial testing phase. Once that phase was completed, a comparable process was used with DTs. There was no suggestion that both classifiers would yield identical results and, indeed, they did not. Rather, the desire was to determine whether or not the results with DTs are comparable to those from NNs. Both classifiers have long histories of being used for diagnosis. As DTs were shown to have comparable results to NNs then it is reasonable to suggest that this approach might be generalizable to other classifiers.

Addressing the Research Questions

Approach to answer Research Question 1: What is an efficient process to identify all acceptable feature sets?

This investigation proceeded from the assumption that removing a feature from a feature set does not materially improve the accuracy of the result obtained (with the possible minor exception of removing noisy features).

Two items were considered in searching the tree. The first is whether a depth-first or a breadth-first search will be used. A tree with an expansion factor of 40 or 50 (not an unusual number of features for a medical diagnosis) would get very wide very quickly (for 50 nodes, potentially over 6,000,000 nodes in the 4^{th} generation and 300,000,000 in the 5^{th}). Hence, breadth-first search was infeasible due to memory requirements. Therefore a depth-first search was employed.

The remainder of this question is dealt with elsewhere in this Methodology section. (See: Method for Searching the Potential Subspace Set.)

Approach to answer Research Question 2: What percentage of the reduced feature sets are above the minimum quality threshold established?

The result from Research Question 1 was a listing of all of the subsets that were at or above the acceptable level. It was a simple matter to count them and to determine the number of potential subsets (2^n-1). The percentage of sets that are acceptable was then immediately calculable. Further, as the sets could be ranked by quality level, the percentages that were satisfactory for any level of performance (above the minimum level specified) would also be trivial to compute.

Approach to answer Research Question 3: What percentage of the qualifying feature sets are on the Pareto-Optimal Frontier?

Answering this question required determining the Pareto-Optimal Frontier (POF) then comparing that with all of the Acceptable Feature Sets (AFS). That is, a subset was on the Pareto-Optimal Frontier or it was not. The calculation was simply the count of those on the frontier divided by the count of all acceptable subsets. This determination was trivial once the AFS and POF had been established. Of note, the AFS required the cost profiles to be generated.

Approach to answer Research Question 4: Does the order in which features are removed have an impact on the number of expansions required?

Considering that a depth-first search was employed, the next issue investigated was the order in which the subsets should be generated. The contribution of a feature to a classifier can be estimated, so the removal order could be most-influential first, least-influential first, or simply random order. The expectation was that the order would be significant and that the lowest training times would be achieved by removing the most-influential feature first and the highest training times will be obtained by removing the least-influential factors first with the random selection falling somewhere in the middle. All three orders were attempted and the results were compared. The expectation related to the ordering of features was generally confirmed.

With respect to most-influential versus least-influential, the determination could have been done once based on the importance when the complete set of features was considered or it could have been determined at each node. It was done once with the full feature set and that ordering was maintained.

Two approaches to determining the importance of individual features were employed. The first, which might be termed 'Simple' was to determine the impact of removing a feature from the complete feature set, determine the result, and then to include only that feature and determine the result. These two values were combined to give a 'Simple' result and the features were then ranked from highest to lowest. Second, the Shapley approach to determining feature importance was employed.

A third approach was considered and abandoned. This third, novel, approach involved determining the AFS at a relatively high level of accuracy (which depended on the levels being used for that particular dataset) then using that information to inform subsequent orderings. Consider a given feature. It might be present in all of the AFS, most of the AFS, or only some of the AFS. It is a simple task to rank the features on that basis of the frequency of their presence and assume that the more frequently a feature was present in the AFS the more important it actually was to the determination. After doing so, that ranking was used when the AFS were generated at lower levels of accuracy. This approach was investigated and found to not offer any material advantage.

The pseudo-code for the DFS is included elsewhere in this Methodology section. (See: Method for Searching the Potential Subspace Set.)

Approach to answer Research Question 5: What is the impact of using a different classifier on the AFS produced and the overall efficiency of the process?

Replacing one classifier with another was done to establish that the process being developed was not tightly tied to the choice of classifier. It is desirable that a user could select a classifier and implement the process with little more effort than making the appropriate calls to the

classifier. The entire process should not need to be regenerated. The swapping of one classifier for another was found to be a straight-forward task with little effort required.

There was no expectation that both classifiers would give identical results. However, both NNs and DTs are frequently used for classification problems so there was some confidence that the results would be comparable. The two items measured were the AFSs produced and the time needed to train and test the classifiers. Rather than measure the clock time, the proxy used will be number of subsets actually tested. The results are detailed in the results section.

Method for Searching the Potential Subset Space

Exhaustively searching the subset space is infeasible for all but the smallest feature sets. Therefore, an alternate method was employed. Assume that a threshold value for minimum acceptable accuracy has been established. The objective was then to find all subsets such that their accuracy with the given classifier was at least the minimum acceptable. Let the subsets be represented by a tree with the root node representing all features present. The nodes represent the models. The nodes can be represented as a pair (F_s, f_p) with F_s as the set of features at the particular node (that is, the subset of features that will be used with the classifier) and f_p is the feature that was removed from the node's parent to generate the node. So, for the root node, F_s would contain all of the features and f_p would be null. This can be represented as $(F, \varnothing)$. Assume that the features have been ranked with a ranking, $R(f)$ denoting the ranking of feature f based on its importance. Then for a given node (F_s, f_p) there exist a set of $|F_s|$ successor nodes. Successor nodes were only generated if the node itself was acceptable (following the assumption that removing a feature does not generally improve the classification). The successor nodes were generated using increasing, decreasing and random order of importance. To generate

the nodes using decreasing order of importance the following was the process. Suppose the initial node is (0, 1, 2, 3, 4)/empty, and the node is acceptable. (The 'empty' signifies that it had no parent.) The (0, 1, 2, 3, 4) asserts that each of these features (0 to 4) are present and could also be written as (11111). The subsets then generated are {(1, 2, 3, 4)/0, (0, 2, 3, 4)/1, (0, 1, 3, 4)/2, (0, 1, 2, 4)/3, (0, 1, 2, 3)/4}. Now consider the subset (1, 2, 3, 4)/0. The '0' value indicates that only index values less than 0 can be removed from this node. As there are none, it has no subsets. However, (0, 2, 3, 4)/1 (alternately 10111/1) has one value less than 1, so its subset would be (2, 3, 4)/0 (alternately 00111/0). This subset would only be generated if the node (0, 2, 3, 4)/1 was acceptable. (As an aside, the notation is not meant to imply the implementation. Rather {(1, 2, 3, 4)/0 …} would be implemented as {(1111,0)…}. The existing notation was selected to improve clarity.)

This generation can be formalized as $\{F_s - \{f\} \mid f \in F_s \wedge R(f) > R(f_p)\}$ if using increasing order of importance or $\{F_s - \{f\} \mid f \in F_s \wedge R(f) < R(f_p)\}$ if using decreasing order of importance. To generate a random (neither consistently increasing nor decreasing) selection order, the features were stored in random order and then searched as if in increasing order.

As each node was examined, it was classified as acceptable or not. If not acceptable, no further action on that node was required. However, if acceptable, the node was added to the acceptable list and its successors are added to the 'to be examined' queue (called priorityQ in the pseudocode below). The pseudocode for increasing order of importance is given as:

```
search(featureSet F, ranking R):
        acceptableModels = [] # acceptable models
        priorityQ = [F] # initialize with root node
        nTrained = 0 # number of models trained
        while priorityQ: # queue not empty
```

```
            nTrained += 1 # train another model
            thisModel = priorityQ.pop() # get first model
            if acceptable(thisModel): # acceptable model
               acceptableModels.append(thisModel) # add to result
               priorityQ.extend(successors(thisModel, R)) # extend priorityQ
return nTrained, acceptableModels

successors(thisModel =(Fs, fp), ranking R):
        models = []
        for f in Fs:
                if R(f) > R(fp): models.append((Fs-{f}, f))
return models
```

Method for Evaluating Approach

The ultimate goal in this exercise was to be able to easily and accurately produce a Pareto-Optimal Frontier using accuracy and cost as the two features to be optimized. Cost profiles were required to establish this frontier. Of note, it had initially been hoped to generate costs that reflected what the real costs might have been. This approach was abandoned for two reasons. First, it was found to be impractical to gather values that would be accurate, given that several components of cost would need to be considered (dollar cost, inconvenience/pain, and potential risk). Second, given that, at best, the cost profile would be an informed guess, their contribution to the results would have been minimal. Further, real cost profiles would be dynamic and individualistic so, even if available, would be of limited utility. Such a cost profile would only be valid for one unique person at one unique time. They would be very expensive (and, hence, infeasible in the context of this study) to obtain. As this investigation was to establish that the method is suitable (or not) synthetic cost values were used.

Cost profiles were artificially generated by the investigator and are asserted to not be accurate, simply to be useful as an example of potential costs. Two approaches to generate synthetic costs were made. One approach was to assume that all costs were identical. The second

approach was to randomly assign costs. These two separate costs functions were used to demonstrate that the assigned costs would have a meaningful impact on the Pareto-Optimal Frontier generated. Specifically, it was not simply the impact of the features but the interaction between costs and accuracy. As a further investigation, specific features were selected and their costs were manipulated to study the impact of such changes.

Given that, at this point, the AFS was established and cost profiles were produced, the Pareto-Optimal Frontier was generated. It was then a trivial matter to select the corresponding trade-off between cost and accuracy. The Pareto-Optimal Frontier was established, and this approach was seen to be successful.

Experimental Setup

The experiments performed followed from the previous discussions. Some of the datasets used could be used directly without any modifications. The Indian Liver dataset had four rows with missing data, so they were simply eliminated. However, they also had skewed results such that it was possible to get answers approximately as good as the classifiers achieved by simply always selecting the dominant choice. Hence, that dataset was trimmed so the resulting reduced set had the same number of positive as negative results. A similar approach was taken with the thyroid dataset, where there were three answers with a significant overweight of one.

MatLab was used as the platform for executing the classifiers. MatLab was cited in many of the literature sources as being the classifier employed. It is a commercial-grade product that has a significant user community and extensive vendor support. The tool generally performed as expected.

Considering the NNs, the first task was to establish a suitable configuration for evaluation. This was done by testing various numbers of nodes and selecting the one that gave the best results. From 25 to 200 nodes were tested, in increments of 25. In the case of a tie, the lower of the two configurations was selected. Gradient descent was the NN method employed. Once the configuration was determined, the full dataset was run 50 times and the results averaged to give the 'nearly best' answer. The term 'nearly best' is used as there may be noisy features present that slightly degrade the answer. Fifty iterations were used so that any unfortunate initial partitioning of the datasets would be averaged away. Once the nearly best results were obtained, then the lower thresholds were established as a percentage of same. For the small datasets, any value might have been used. If there are only 10 features, there are, at most 1023 combinations of interest (it is assumed the empty feature set offers no information, so 2^10 – 1). However, some of the larger datasets had evaluation counts in the millions, even when the threshold was at 99.5% of the answer obtained using all features. This high number was caused two factors: there were a large number of AFS (hence a large number of subsets of same) and the search order was increasing (which was eventually confirmed to be the most expensive order). So, had all of the datasets been run at the original planned levels of 99, 96 and 93% accuracy, the execution times would have been unacceptable. Nevertheless, it was possible to observe that the size of the AFS increased as the threshold level was decreased.

With the configuration of the classifier and the threshold level established, the code to automate the execution was developed. The pseudo-code has been described elsewhere in this document. It required little more than tying in to the classifier for the evaluation and writing the results (that is, acceptable subsets with respective accuracy obtained) to file for later processing.

Once the code was developed and executed, the results consisted of a collection of subsets and accuracies. The Pareto-Optimal Frontier was established using two sets of synthetic costs. The Pareto-Optimal optimization was for cost (lower preferred) and accuracy (higher preferred).

Once the experiment had been completed using NNs, it was repeated with DTs. The results obtained were then compared. They were not identical. They are described in the results section.

Data Sets Used

The datasets used were either synthetically generated or from public sources. No collection of original data was done. The small datasets were used to establish that the overall approach was sound. Then the larger datasets were used to further explore the research questions.

A total of five datasets were used. There were two synthetic datasets that the author generated primarily to establish that the process was correct, although they also provided some additional information. A third dataset was sourced from the UCI catalog related to Indian Liver Patient Disease (ILPD). The ILPD consists of 10 attributes and 583 data rows. It is sufficiently small to establish that the overall approach is sound. It is available at http://archive.ics.uci.edu/ml/datasets/ILPD+%28Indian+Liver+Patient+Dataset%29. (Acknowledgement: Ramana, B., Babu, M., and Venkateswarlu, N. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.)

Also from the UCI resource, data was used with respect to the Wisconsin Breast (Diagnostic) Cancer. The Breast Cancer Wisconsin (Diagnostic) Data Set is available at http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29.

(Acknowledgement: Wolberg, W., Street, W., and Mangasarian, O. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.) It consists of 30 attributes and has 569 data rows. It is sufficiently large to establish that the process will scale. MatLab has many built in datasets. Their thyroid dataset was also used, primarily as confirmation of the some of the other observations. The thyroid dataset had 21 features and 498 rows data rows were used.

Summary

The data required was obtained from the sources noted. The features were ranked using two different methods (Simple and Shapley), with Novel Present Count found to be not of advantage. Two types of classifiers were used. Both have been shown to be widely used as classifiers generally and for medical diagnoses in particular. A threshold value for each dataset was established. Configurations were established for the classifiers. The evaluations of each classifier were executed on MatLab, which is a platform that is commonly referenced in the literature as a platform of choice and common usage. The DFS has been described above using increasing, decreasing, and random order of importance of features. The code to run the DFS and interface with the MatLab classifier was generated and executed. Removal in the decreasing order was materially better than random which was materially better than increasing order. The execution of the code produced a set of acceptable subsets of features. Then, using both unit and random cost profiles, it was a trivial matter to determine the Pareto-Optimal Frontiers. The process was repeated for the DTs and the results are presented below.

Chapter 4

Results

Brief Description of the Overall Organization of the Results Section

Five datasets were used; two synthetic and three natural ones. They were labeled Synth (for synthetic), S 2 (for synthetic 2), IL (for Indian Liver), Thyroid (for Thyroid) and WBC (for Wisconsin Breast Cancer). The two classifiers used were Neural Networks (NN) and Decision Trees (DT). The environment used was MatLab (Version R2017a). The Neural Network used was gradient descent. For each dataset, the Neural Network was tested as to which number of nodes produced the optimal result using all of the features and that configuration was continued throughout for that particular dataset. The Decision Tree was used with moderate pruning. For each dataset and each classifier, a 'best' result was obtained using the full dataset.

Table 1

The Best Value Obtained Using the Complete Data Set for Each Classifier

Best Value	NN	DT	Features	Observations	Classes
Synth	1	0.8728	10	1000	2
S 2	0.8947	0.7566	13	1000	2
IL	0.72	0.638	10	330	2
Thyroid	0.8919	0.993	21	498	3

WBC	0.9858	0.9245	30	269	2

For each classifier and dataset, three different threshold levels were used. For Synth, S 2 and IL the levels were 0.99, 0.95 and 0.93. For Thyroid and WBC the levels were 0.999, 0.997 and 0.995. These choices were necessarily arbitrary but gave a sufficiently large set of results without being excessive. Two methods of ordering the features were used, with Simple (described elsewhere in this paper) and the well-known Shapley ordering. Three approaches were taken to feature elimination: increasing order, decreasing order and random order.

Many of the tables presented will follow a similar style as to the one following. The following presents the first few rows from the first synthetic dataset and is used simply to clarify some of the details that will appear later:

Table 2

Example Table

ORDERING SHAPLEY	cAFS	cTST	cFST	Ratio cAFS/cTST	cPOF @ unit cost	cPOF/cAFS @ unit cost	cPOF @ random cost	cPOF/cAFS @ random cost
Synth INC @ 0.93 NNs	32	192	160	0.167	6	0.188	2	0.063
Synth INC @ 0.96 NNs	32	192	160	0.167	1	0.031	1	0.031

The upper left hand cell indicates that Shapley ordering was used to generate this set of results. The remainder of the left hand column displays the particular configuration used to generate the data. This information is displayed as <dataset used> <order of feature removal> @ <threshold level > <the classifier used, NNs for Neural Networks and DTs for Decision Trees>. The threshold level is measured against the value obtained when using the complete dataset with

the classifier noted. The column headers are as follows. The count of acceptable feature sets is referred to as count of AFS (cAFS). The count of total sets tested is count of TST (cTST). The count of failed sets tested is count of FST (cFST). The count of sets on the Pareto-Optimal Frontier is count of POF (cPOF).

Addressing the Research Questions

Answer to Research Question 1: What is an efficient process to identify all acceptable feature sets?

There were several assumptions made with respect to this question and they are well documented elsewhere in this paper, so will not be restated (see Chapter 1: Assumptions, Limitations and Delimitations). Working within those assumptions, the main issue is whether or not an efficient method of finding the AFS was developed. One major interest was to discover an efficient process. Efficiency can be approximated by the ratio of the count of Acceptable Feature Subsets to the count of the Total number of subsets tested. Let this be called the Efficiency Ratio. (Of note, one could also have used the count of Acceptable Feature Subsets to the count of Failed Subsets tested; for purposes of consistency and brevity, only the Efficiency Ratio will be discussed.) A higher Efficiency Ratio (that is, nearer to one) is deemed better in that fewer non-acceptable subsets were tested relative to the number of acceptable subsets. A lower Efficiency Ratio (that is, nearer to zero) is deemed worse in that more subsets were tested relative to the number of acceptable subsets.

For both NNs and DTs, the ratio of Efficiency Ratio was almost always larger (better) when features were removed in decreasing order of significance and almost always smaller (worse) when removed in increasing order of significance. The only exceptions were when the

number of features was very small. For all of the natural datasets, the results described above were consistently the case.

Two examples will be used to highlight this result.

Considering S 2, the following table compares the results for Simple Ordering.

Table 3

S 2 using Simple Ordering

Ordering Simple	cAFS	cTST	cFST	Ratio cAFS / cTST	cPOF @ unit cost	Ratio cPOF / cAFS @ unit cost	cPOF @ random cost	Ratio cPOF / cAFS @ random cost
S 2 INC @ 0.93 NNs	73	312	239	0.234	3	0.041	5	0.068
S 2 INC @ 0.96 NNs	39	291	253	0.134	3	0.077	5	0.128
S 2 INC @ 0.99 NNs	31	280	249	0.111	3	0.097	5	0.161
S 2 DEC @ 0.93 NNs	51	84	33	0.607	2	0.039	3	0.059
S 2 DEC @ 0.96 NNs	31	40	9	0.775	3	0.097	4	0.129
S 2 DEC @ 0.99 NNs	8	18	10	0.444	3	0.375	3	0.375
S 2 RAN @ 0.93 NNs	59	175	127	0.337	2	0.034	4	0.068
S 2 RAN @ 0.96 NNs	31	127	96	0.244	2	0.065	3	0.097
S 2 RAN @ 0.99 NNs	14	59	45	0.237	1	0.071	1	0.071
S 2 INC @ 0.93 DTs	1303	2176	873	0.599	6	0.005	14	0.011
S 2 INC @ 0.96 DTs	541	966	425	0.56	3	0.006	7	0.013
S 2 INC @ 0.99 DTs	112	296	184	0.378	4	0.036	7	0.063
S 2 DEC @ 0.93 DTs	222	250	28	0.888	4	0.018	8	0.036
S 2 DEC @ 0.96 DTs	90	100	10	0.9	2	0.022	3	0.033
S 2 DEC @ 0.99 DTs	29	36	7	0.806	3	0.103	4	0.138
S 2 RAN @ 0.93 DTs	1165	2133	968	0.546	5	0.004	9	0.008

S 2 RAN @ 0.96 DTs	459	956	497	0.48	5	0.011	5	0.011
S 2 RAN @ 0.99 DTs	86	265	179	0.325	2	0.023	2	0.023

Note that the Efficiency Ratio is consistently higher for Decreasing order of removal than for Increasing with Random being the intermediate value. This is highlighted in the segment of Table 3 presented in Table 4.

Considering only the Efficiency Ratio for the values at the 0.93 threshold level, the following values are observed.

Table 4

The Efficiency Ratio for Selected Values

Configuration	Efficiency Ratio
S 2 INC @ 0.93 NNs	0.234
S 2 DEC @ 0.93 NNs	0.607
S 2 RAN @ 0.93 NNs	0.337
S 2 INC @ 0.93 DTs	0.599
S 2 DEC @ 0.93 DTs	0.888
S 2 RAN @ 0.93 DTs	0.546

The values in Table 4 relate to the 0.93 Threshold Level for S 2 relative to Table 3 (above). For both NNs and DTs the decreasing order produces the best (greatest) Efficiency Ratio. Of note, the relative importance of features was completely known for this dataset as it was a constructed (synthetic) set.

Considering the same comparison for the Thyroid dataset using the Shapley ordering, the following results are obtained (note that the Random results were taken from the Simple ordering results because random ordering is just that: random):

Table 5

The Efficiency Ratios for Selected Thyroid Configurations

Configuration	Efficiency Ratio
Thyroid INC @ 0.995 NNs	0.685
Thyroid DEC @ 0.995 NNs	0.656
Thyroid RAN @ 0.995 NNs	0.808
Thyroid INC @ 0.995 DTs	0.286
Thyroid DEC @ 0.995 DTs	0.999
Thyroid RAN @ 0.995 DTs	0.666

Although the results support the notion that Decreasing order is superior to Random which is superior to Increasing order for the DTs, that conclusion is not supported by the NNs where the best observed is the Random order while the worst is Decreasing, although there is only a modest difference between Increasing and Decreasing.

The following table summarizes the cAFS across all datasets.

Table 6

Summarized Efficiency Ratios across all Data Sets

	Inc	Dec	Ran
Synth Simple	0.216	0.863	0.614
Synth Shapley	0.183	0.755	0.614
S 2 Simple	0.336	0.737	0.362
S 2 Shapley	0.211	0.694	0.362
IL Simple	0.559	0.6	0.539
IL Shapley	0.487	0.555	0.539
Thyroid Simple	0.447	0.844	0.703
Thyroid Shapley	0.469	0.779	0.703
WBC Simple	0.371	0.374	0.438
WBC Shapley	0.311	0.339	0.438
Just Synthetics	0.236	0.762	0.488
Just Natural	0.441	0.582	0.539
Overall	**0.359**	**0.654**	**0.520**

It can be noted that Decreasing is consistently superior to Random which is consistently superior to Increasing. So while there may be specific instances where this general rule does not hold, they are the exceptions. It is reasonable to conclude that selection order is material in achieving an efficient selection process.

Therefore, an efficient process can be described as follows. Order the features using Shapley ordering. Next, select an appropriate threshold value (this will require experimentation and may require consultation with an end-user group). Search the space using the decreasing order of removal. Using some cost function, prepare the POF.

Answer to Research Question 2: What percentage of the reduced feature sets are above the minimum quality threshold established?

There are $2^n - 1$ potential feature sets (cPDS) when there are n features. Let the ratio of cAFS to cPDS be termed the Acceptable-Potential Ratio. For most of the datasets examined, the count of AFS increased as the minimum quality threshold was decreased. The only exceptions were with the synthetic sets where the counts were frequently identical – that is, decreasing the threshold had no effect over the range used. A representative sampling of the results is summarized in the table below. As the Random order always produced an intermediate result they are omitted.

Table 7

Representative Listing of the Acceptable-Potential Ratio

ORDERING SHAPLEY	cAFS	n	cPDS	Acceptable-Potential Ratio
Synth INC @ 0.93 NNs	32	10	1023	0.03128055
Synth INC @ 0.96 NNs	32	10	1023	0.03128055

Synth INC @ 0.99 NNs	32	10	1023	0.03128055
Synth DEC @ 0.93 NNs	32	10	1023	0.03128055
Synth DEC @ 0.96 NNs	32	10	1023	0.03128055
Synth DEC @ 0.99 NNs	31	10	1023	0.03030303
Synth INC @ 0.93 DTs	150	10	1023	0.14662757
Synth INC @ 0.96 DTs	98	10	1023	0.09579668
Synth INC @ 0.99 DTs	1	10	1023	0.00097752
Synth DEC @ 0.93 DTs	96	10	1023	0.09384164
Synth DEC @ 0.96 DTs	82	10	1023	0.0801564
Synth DEC @ 0.99 DTs	1	10	1023	0.00097752
S 2 INC @ 0.93 NNs	78	13	8191	0.00952265
S 2 INC @ 0.96 NNs	39	13	8191	0.00476132
S 2 INC @ 0.99 NNs	27	13	8191	0.0032963
S 2 DEC @ 0.93 NNs	45	13	8191	0.00549383
S 2 DEC @ 0.96 NNs	36	13	8191	0.00439507
S 2 DEC @ 0.99 NNs	20	13	8191	0.0024417
S 2 INC @ 0.93 DTs	1345	13	8191	0.16420461
S 2 INC @ 0.96 DTs	573	13	8191	0.06995483
S 2 INC @ 0.99 DTs	59	13	8191	0.00720303
S 2 DEC @ 0.93 DTs	684	13	8191	0.08350629
S 2 DEC @ 0.96 DTs	142	13	8191	0.0173361
S 2 DEC @ 0.99 DTs	45	13	8191	0.00549383
IL INC @ 0.93 NNs	124	10	1023	0.12121212
IL INC @ 0.96 NNs	18	10	1023	0.01759531
IL INC @ 0.99 NNs	5	10	1023	0.00488759
IL DEC @ 0.93 NNs	343	10	1023	0.33528837
IL DEC @ 0.96 NNs	30	10	1023	0.02932551
IL DEC @ 0.99 NNs	4	10	1023	0.00391007
IL INC @ 0.93 DTs	721	10	1023	0.70478983
IL INC @ 0.96 DTs	299	10	1023	0.29227761
IL INC @ 0.99 DTs	18	10	1023	0.01759531
IL DEC @ 0.93 DTs	577	10	1023	0.56402737
IL DEC @ 0.96 DTs	208	10	1023	0.20332356
IL DEC @ 0.99 DTs	17	10	1023	0.01661779
Thyroid INC @ 0.995 NNs	27547	21	2097151	0.01313544
Thyroid INC @ 0.997 NNs	8126	21	2097151	0.00387478
Thyroid INC @ 0.999 NNs	9981	21	2097151	0.00475931
Thyroid DEC @ 0.995 NNs	2275	21	2097151	0.00108481
Thyroid DEC @ 0.997 NNs	1125	21	2097151	0.00053644
Thyroid DEC @ 0.999 NNs	805	21	2097151	0.00038385
Thyroid INC @ 0.995 DTs	266070	21	2097151	0.12687212
Thyroid INC @ 0.997 DTs	247268	21	2097151	0.11790663
Thyroid INC @ 0.999 DTs	37262	21	2097151	0.01776791

Thyroid DEC @ 0.995 DTs	259147	21	2097151	0.12357098
Thyroid DEC @ 0.997 DTs	205213	21	2097151	0.09785323
Thyroid DEC @ 0.999 DTs	43229	21	2097151	0.0206132
WBC INC @ 0.995 NNs	330	30	1073741823	0.00000031
WBC INC @ 0.997 NNs	15	30	1073741823	0.00000001
WBC INC @ 0.999 NNs	21	30	1073741823	0.00000002
WBC DEC @ 0.995 NNs	573	30	1073741823	0.00000053
WBC DEC @ 0.997 NNs	15	30	1073741823	0.00000001
WBC DEC @ 0.999 NNs	5	30	1073741823	0.000000005
WBC INC @ 0.995 DTs	135993	30	1073741823	0.00012665
WBC INC @ 0.997 DTs	166980	30	1073741823	0.00015551
WBC INC @ 0.999 DTs	178	30	1073741823	0.00000017
WBC DEC @ 0.995 DTs	691134	30	1073741823	0.00064367
WBC DEC @ 0.997 DTs	28536	30	1073741823	0.00002658
WBC DEC @ 0.999 DTs	316	30	1073741823	0.00000029

From the above table, it can be observed that a relatively small fraction of the potential datasets are acceptable and that count is related closely to the acceptable threshold employed and the number of features in the dataset. As the acceptable threshold is raised, fewer sets were acceptable. As the number of features increased, the count of potential subsets quickly rose. From the above table, the maximum ratio Acceptable-Potential Ratio observed was 0.7047 with IL INC @ 0.93 DTs. The minimum observed was 0.000000005 for WBC DEC @ 0.999 NNs. The average value was 0.0637.

From the data presented, it can be observed that the Acceptable-Potential Ratio decreases as the number of features increases and the Acceptable-Potential Ratio increases as the threshold level decreases. While the cPDS is fixed by number of features, the cAFS is dependent on the necessarily arbitrary threshold level. In any real world scenario, it is probably desirable to keep the threshold level relatively high but, for academic purposes, it could be set anywhere. Therefore, it would be possible to have essentially any number of AFS.

Answer to Research Question 3: What percentage of the qualifying feature sets are on the Pareto-Optimal Frontier?

The Pareto-Optimal Frontier is used to explore the tension between accuracy and cost. One underlying assumption was that, given unlimited and free resources, the best (or nearly best) results could be obtained by simply executing all tests with the desired classifier. One key feature of this investigation was to find results that are nearly as good as the best obtainable but at a substantially lower cost. To that end, costs must be provided. Two sets of costs were used. The first was unit cost where every test was assessed the same cost (unit cost). This is not a realistic assumption and is not held forth to be so. However, it is convenient when the effort of determining the actual cost would be excessively problematic. Further, it can be used to demonstrate whether or not the elimination of some tests (regardless of their cost) has limited negative effect on the outcome but some impact on the cost. The second type of cost employed was a random cost. Again, this is not held out to be a realistic representation of the actual cost. Rather, the objective was to determine if, as costs changed, the Acceptable Feature Sets on the Pareto-Optimal Frontier might also change. As further experiment, some of the unit costs were altered (that is, were changed from being unit to something else) to determine the impact of repricing specific features. Let the ratio of cPOF/cAFS be called the on-Frontier ratio.

The following table summarizes the results using Shapley ordering and ignoring the Random selection ordering.

Table 8

The on-Frontier Ratio Across Selected Configurations

ORDERING SHAPLEY	cAFS	cPOF@ unit cost	cPOF / cAFS @ unit cost	cPOF@ random cost	cPOF / cAFS @ random cost
Synth INC @ 0.93 NNs	32	6	0.188	2	0.063
Synth INC @ 0.96 NNs	32	1	0.031	1	0.031
Synth INC @ 0.99 NNs	32	1	0.031	1	0.031
Synth DEC @ 0.93 NNs	32	1	0.031	1	0.031
Synth DEC @ 0.96 NNs	32	1	0.031	1	0.031
Synth DEC @ 0.99 NNs	31	1	0.032	1	0.032
Synth INC @ 0.93 DTs	150	2	0.013	2	0.013
Synth INC @ 0.96 DTs	98	3	0.031	3	0.031
Synth INC @ 0.99 DTs	1	1	1	1	1
Synth DEC @ 0.93 DTs	96	1	0.01	2	0.021
Synth DEC @ 0.96 DTs	82	1	0.012	1	0.012
Synth DEC @ 0.99 DTs	1	1	1	1	1
S 2 INC @0.93 NNs	78	7	0.09	7	0.09
S 2 INC @0.96 NNs	39	4	0.103	4	0.103
S 2 INC @0.99 NNs	27	3	0.111	2	0.074
S 2 DEC @0.93 NNs	45	2	0.044	2	0.044
S 2 DEC @0.96 NNs	36	2	0.056	2	0.056
S 2 DEC @0.99 NNs	20	2	0.1	2	0.1
S 2 INC @0.93 DTs	1345	6	0.004	12	0.009
S 2 INC @0.96 DTs	573	4	0.007	8	0.014
S 2 INC @0.99 DTs	59	3	0.051	4	0.068
S 2 DEC @0.93 DTs	684	4	0.006	6	0.009
S 2 DEC @0.96 DTs	142	2	0.014	2	0.014
S 2 DEC @0.99 DTs	45	2	0.044	4	0.089
IL INC @ 0.93 NNs	124	4	0.032	5	0.04
IL INC @ 0.96 NNs	18	2	0.111	2	0.111
IL INC @ 0.99 NNs	5	2	0.4	2	0.4
IL DEC @ 0.93 NNs	343	5	0.015	10	0.029
IL DEC @ 0.96 NNs	30	2	0.067	2	0.067
IL DEC @ 0.99 NNs	4	3	0.75	2	0.5
IL INC @ 0.93 DTs	721	3	0.004	5	0.007
IL INC @ 0.96 DTs	299	4	0.013	10	0.033
IL INC @ 0.99 DTs	18	2	0.111	3	0.167
IL DEC @ 0.93 DTs	577	4	0.007	5	0.009

IL DEC @ 0.96 DTs	208	5	0.024	8	0.038
IL DEC @ 0.99 DTs	17	3	0.176	3	0.176
Thyroid INC @ 0.995 NNs	27547	3	0.000109	6	0.000218
Thyroid INC @ 0.997 NNs	8126	2	0.000246	6	0.000738
Thyroid INC @ 0.999 NNs	9981	5	0.000501	5	0.000501
Thyroid DEC @ 0.995 NNs	2275	4	0.001758	4	0.001758
Thyroid DEC @ 0.997 NNs	1125	4	0.003556	7	0.006222
Thyroid DEC @ 0.999 NNs	805	6	0.007453	6	0.007453
Thyroid INC @ 0.995 DTs	266070	5	0.000019	2	0.000008
Thyroid INC @ 0.997 DTs	247268	1	0.000004	1	0.000004
Thyroid INC @ 0.999 DTs	37262	1	0.000027	3	0.000081
Thyroid DEC @ 0.995 DTs	259147	3	0.000012	11	0.000042
Thyroid DEC @ 0.997 DTs	205213	2	0.00001	1	0.000005
Thyroid DEC @ 0.999 DTs	43229	4	0.000093	2	0.000046
WBC INC @ 0.995 NNs	330	10	0.030303	6	0.018182
WBC INC @ 0.997 NNs	15	2	0.133333	2	0.133333
WBC INC @ 0.999 NNs	21	3	0.142857	4	0.190476
WBC DEC @ 0.995 NNs	573	2	0.00349	3	0.005236
WBC DEC @ 0.997 NNs	15	1	0.066667	1	0.066667
WBC DEC @ 0.999 NNs	5	1	0.2	1	0.2
WBC INC @ 0.995 DTs	135993	9	0.000066	11	0.000081
WBC INC @ 0.997 DTs	166980	7	0.000042	11	0.000066
WBC INC @ 0.999 DTs	178	3	0.016854	7	0.039326
WBC DEC @ 0.995 DTs	691134	4	0.000006	10	0.000014
WBC DEC @ 0.997 DTs	28536	9	0.000315	4	0.00014
WBC DEC @ 0.999 DTs	316	4	0.012658	5	0.015823

The average value the on-Frontier ratio at unit cost is 0.090 and 0.087 for random cost. The minimum value observed was 0.000004 for both unit and random cost while the maximum value for both unit and random cost was 1 occurring where there was only 1 value acceptable. The minimum count of acceptable sets on the POF is 1 for both unit and random costing while the respective maximums are 10 and 12. The average count is 3.25 for unit costing and 4.133 for random costing.

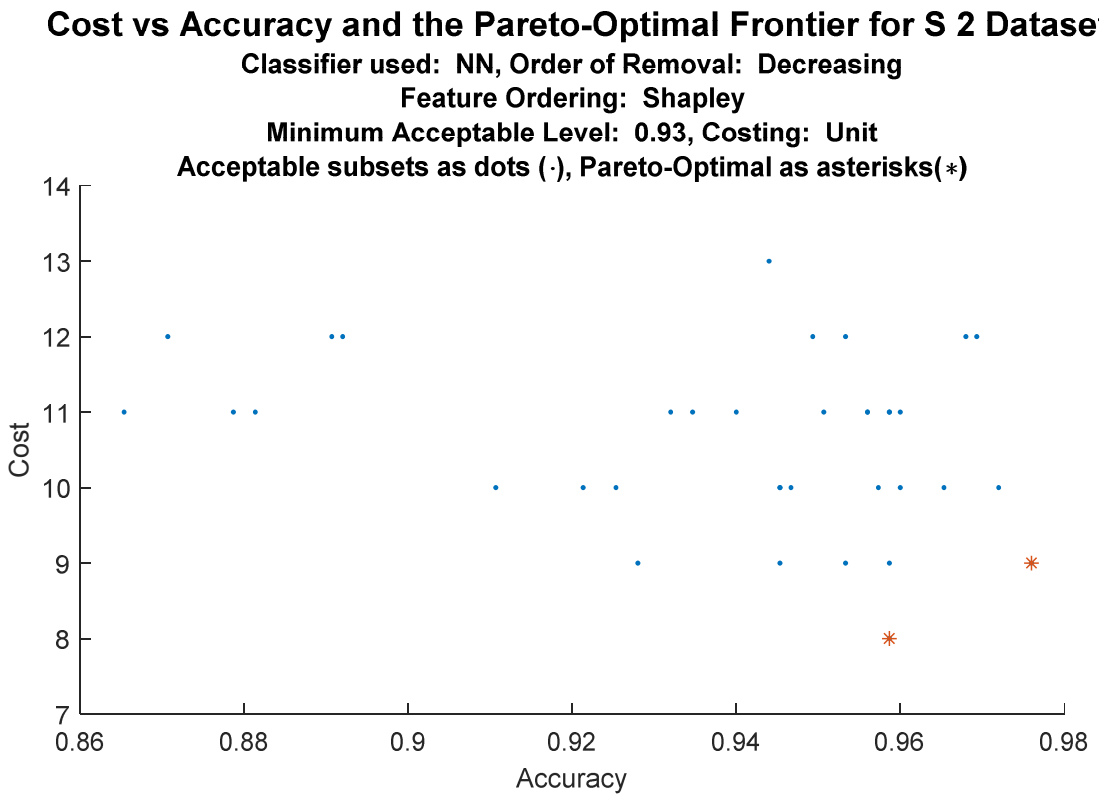

Figure 1. Pareto-Optimal Frontier for S 2 the Dataset.

The above graph displays the POF for S 2 using NNs, decreasing order of feature removal, Shapley feature ordering, 0.93 as the minimum acceptable level, and unit costing.

Table 9

Cost Versus Features Used for POF for S 2

Accuracy Value	Cost	Features Used
0.95867	8	0000011111111
0.976	9	0001011111111

The above table displays cost versus features used for POF for S 2 using NNs, with decreasing order of feature removal, with Shapley feature ordering, with 0.93 as the minimum acceptable level, and unit costing.

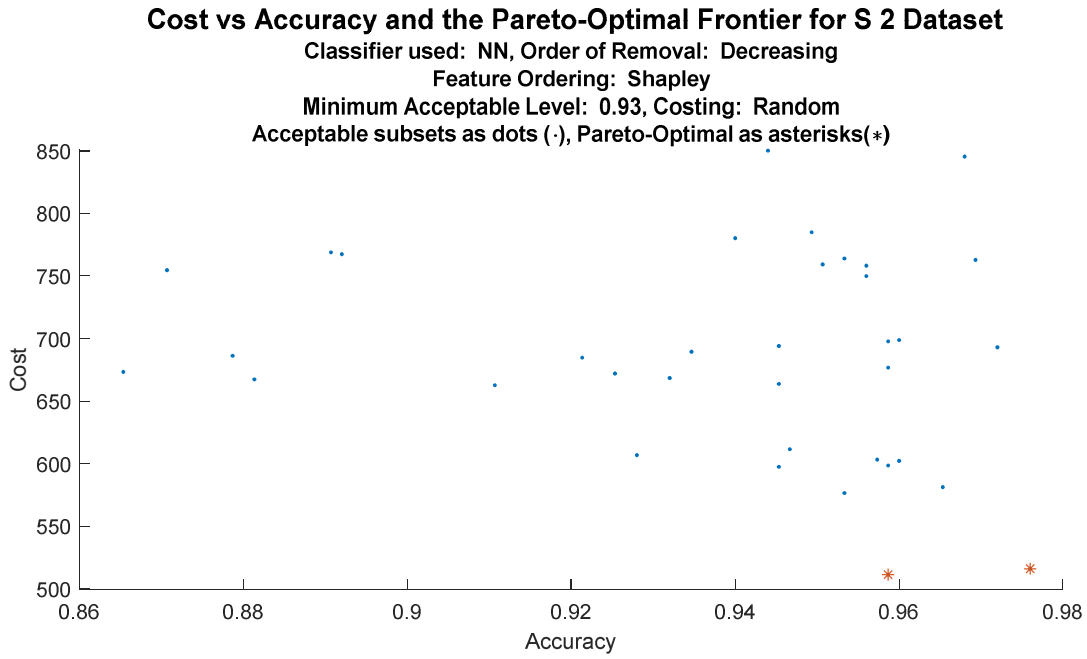

Figure 2. Pareto-Optimal Frontier for the S 2 Dataset.

The above graph displays cost versus accuracy for the POF for S 2 using NNs, with decreasing order of removal, with Shapley feature ordering, with minimum acceptable level of 0.93 and random costing.

Table 10

Cost Versus Features Used for POF for S 2

Accuracy Value	Cost	Features Used
0.95867	511	0000011111111
0.976	516	0001011111111

The above table displays the cost versus features used for POF for S 2 using NNs, with decreasing order of removal, with Shapley feature ordering, with the minimum acceptable level of 0.93 and with random costing.

The count of AFS was 36 for the above two examples (unit and random costing (note: the count of AFS is always identical for unit and random costing, however the count of the POF may

vary)). The above two representations vary only in the approach to cost. In this case, the same two subsets of features were on the Pareto-Optimal Frontier, simply with different cost values. It can be observed that either four or five of the features can be eliminated, depending on which set is desired. Further, the overall accuracy obtained when all features were present was 0.89467, while all of the subsets on the Pareto-Optimal Frontier produced superior answers. So not only are fewer features required, expressly implying a lower cost, but a better result was obtained. A superior answer at a lower cost is always to be desired.

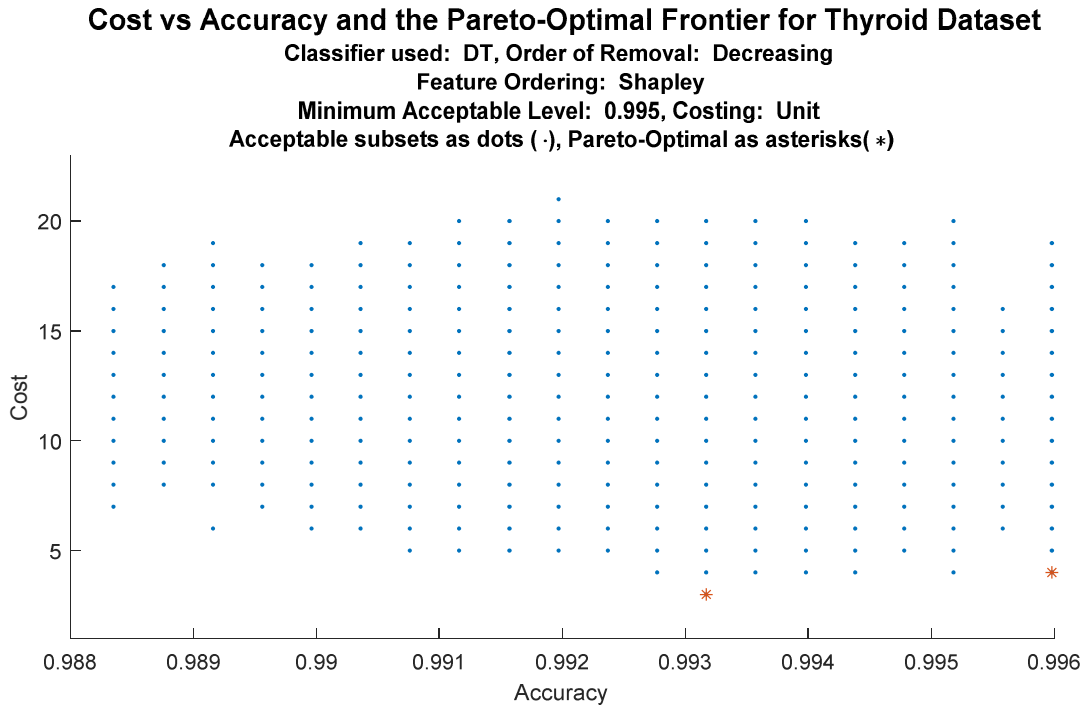

Figure 3. Pareto-Optimal Frontier for Thyroid Dataset.

The above figure displays the Pareto-Optimal Frontier for the Thyroid dataset using DTs, with decreasing order of removal, using Shapley feature ordering, with 0.995 as the minimum acceptable level and with unit costing.

Table 11

Cost Versus Features Used for POF for the Thyroid Dataset

Accuracy Value	Cost	Features Used
0.99317	3	00000000000000001011
0.99598	4	00100000000000001011

The above table displays the accuracy, cost and features used for the Thyroid dataset using DTs, with decreasing order of removal, with Shapley feature ordering, with 0.995 as the minimum acceptable level and with unit costing. Compare the results of the above table and feature using unit costing with the results obtained for random costing, presented below.

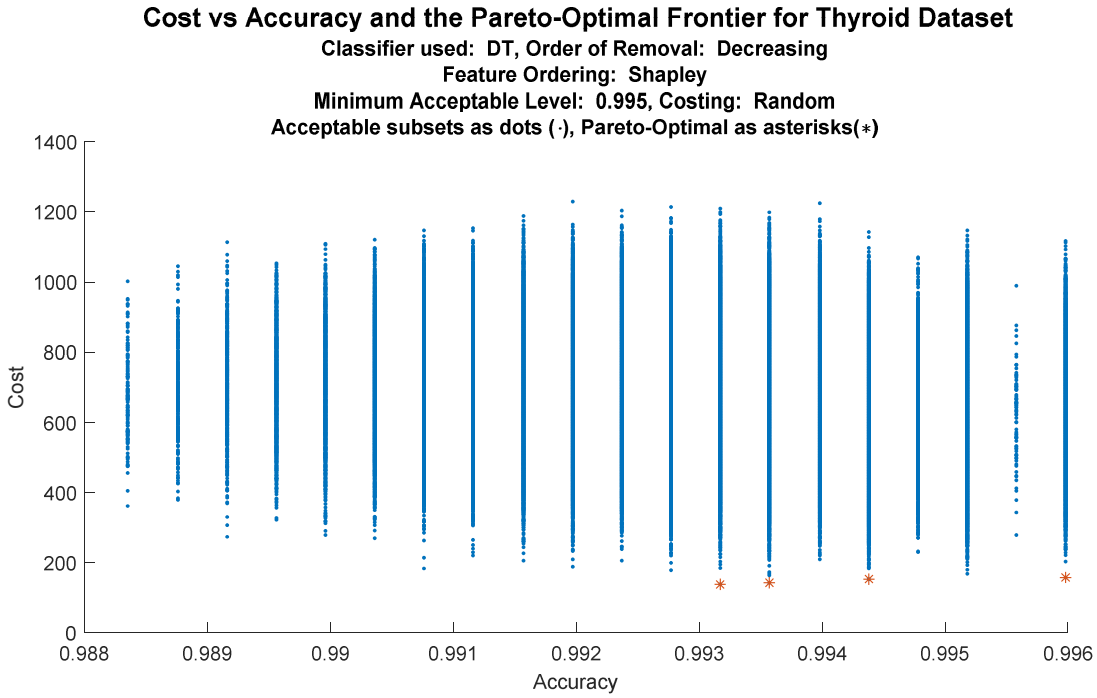

Figure 4. Pareto-Optimal Frontier for the Thyroid Dataset.

The above figure displays the Pareto-Optimal Frontier for Thyroid dataset using DTs, with decreasing order of removal, using Shapley feature ordering, with 0.995 as the minimum acceptable level, and with random costing.

Table 12

Cost Versus Features Used for POF for the Thyroid Dataset

Accuracy Value	Cost	Features Used
0.99317	137	00000000000000001011
0.99357	142	00010000000000001011
0.99438	152	00000010000000001011
0.99598	157	00010010000000001011

The above table displays the accuracy, cost and features used for the Thyroid dataset using DTs, with decreasing order of removal, with Shapley feature ordering, with 0.995 as the minimum acceptable level and with random costing. For the Thyroid dataset using DTs with all features present the best overall accuracy obtained was 0.99297. There were 259147 AFSs observed at the 0.995 level of accuracy. For all of the datasets on the Pareto-Optimal Frontier (the two from unit costing and four from random costing) the results obtained were slightly better (than the best overall at 0.99297) and the costing was very much lower. Again, the elimination of some features has not only lowered the cost but improved the overall answer, the best being 0.99598 with unit cost of 4 or random cost of 157.

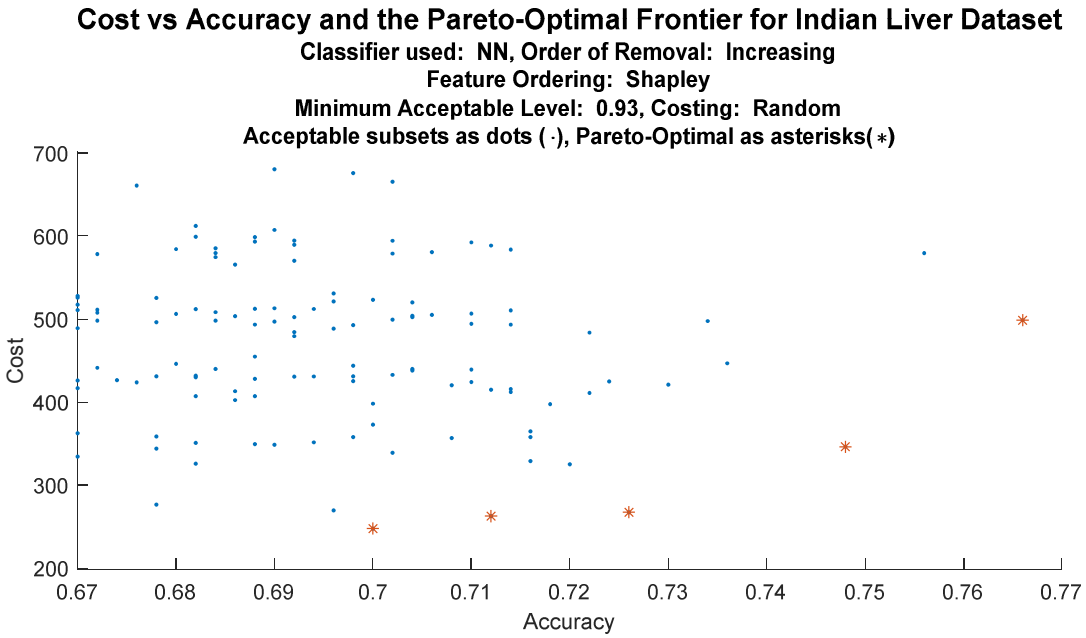

Figure 5. Pareto-Optimal Frontier for the Indian Liver Dataset.

The above figure displays the Pareto-Optimal Frontier for the Indian liver dataset using NNs, with increasing order of feature removal, using Shapley feature ordering, with 0.93 as the minimum acceptable level, and with random costing.

Table 13

Cost Versus Features Used for POF for the Thyroid Dataset

Accuracy Value	Cost	Features Used
0.7	248	1110000000
0.712	263	1110000100
0.726	268	1111000100
0.748	346	1000110101
0.766	499	1110110101

The above table displays the accuracy, cost and features used for the Indian Liver dataset using the NN classifier, with increasing order of removal, with Shapley feature ordering, with 0.93 as the minimum acceptable level and with random costing.

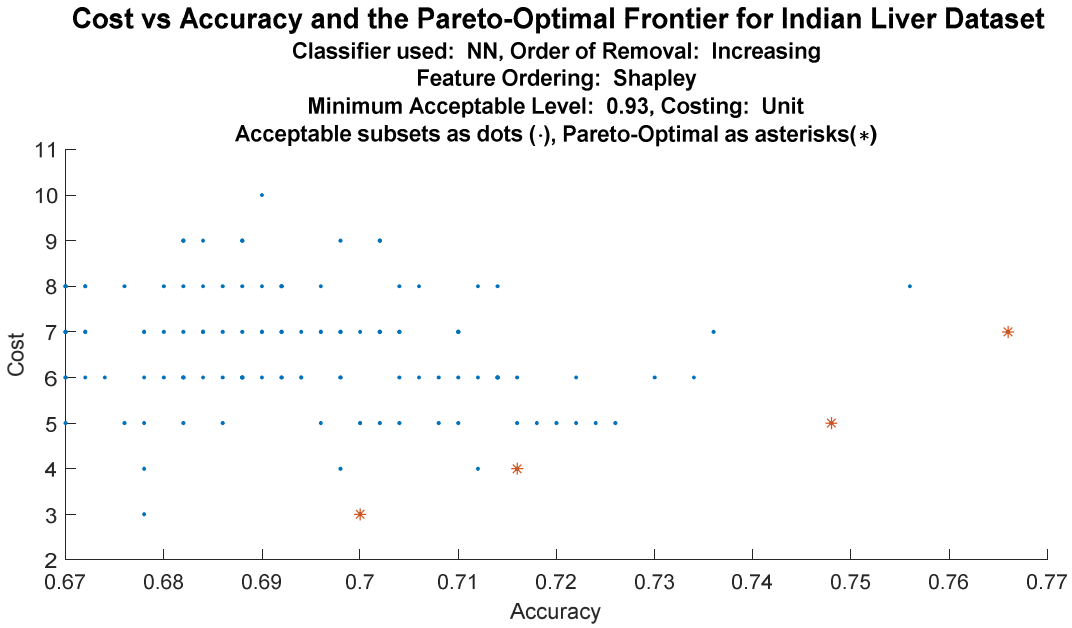

Figure 6. Pareto-Optimal Frontier for the Indian Liver Dataset.

The above figure displays the Pareto-Optimal Frontier for the Indian Liver dataset using the NN classifier, with increasing order of removal, using Shapley feature ordering, with 0.93 as the minimum acceptable level, and with unit costing.

Table 14

Accuracy, Cost and Features Used for POF for the Indian Liver Dataset

Accuracy Value	Cost	Features Used
0.7	3	1110000000
0.716	4	1110000001
0.748	5	1000110101
0.766	7	1110110101

The above table displays the accuracy, cost and features used for the Indian Liver dataset using the NN classifier, with increasing order of removal, with Shapley feature ordering, with 0.93 as the minimum acceptable level and with unit costing. The above graphs represent the results for unit and random costing for the Indian Liver dataset at the 0.93 acceptance level. The maximum

observed accuracy when all features were present was 0.7128. Three of Pareto-Optimal sets with random costing and two of them with unit costing were actually above the 'all-features' level of 0.7128. In all cases, at least three of the features were not required. In the most extreme case, only 3 of the 10 features were required to yield an accuracy of 0.7 (versus the all-feature one of 0.7128). In this case the worst sets on the Pareto-Optimal Frontier were only slightly worse than the all-features value and then at a much lower cost. Those with superior performance were produced at a lower cost than the all-feature instance. The count of AFS for this configuration was 124.

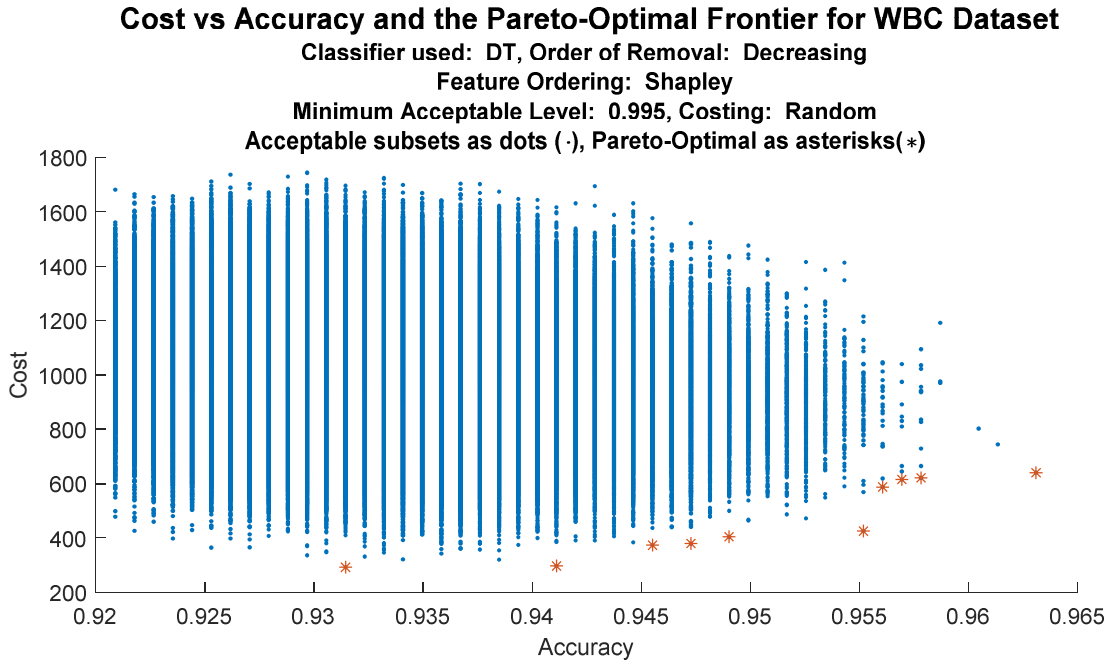

Figure 7. Pareto-Optimal Frontier for the WBC Dataset.

The above figure displays the Pareto-Optimal Frontier for the WBC dataset using the DT classifier, with decreasing order of feature removal, using Shapley feature ordering, with 0.995 as the minimum acceptable level, and with random costing.

Table 15

Accuracy, Cost and Features Used for POF for the WBC Dataset

Accuracy Value	Cost	Features Used
0.93146	292	0000000100000000011010101010
0.94112	297	0001000100000000011010101010
0.94552	373	0000000101000000011010101010
0.94728	379	0001000100001000011010101010
0.94903	404	0000000100010100011110101010
0.95518	425	0000000100000000110011110001010
0.95606	587	0010000010000001111111001101010
0.95694	615	0011000000010000011001110111010
0.95782	621	0011010010000100101101100010 10
0.96309	640	0010010010010000010011010110 10

The above table displays the accuracy, cost and features used for the WBC dataset using the DT classifier, with decreasing order of removal, with Shapley feature ordering, with 0.995 as the minimum acceptable level and with random costing.

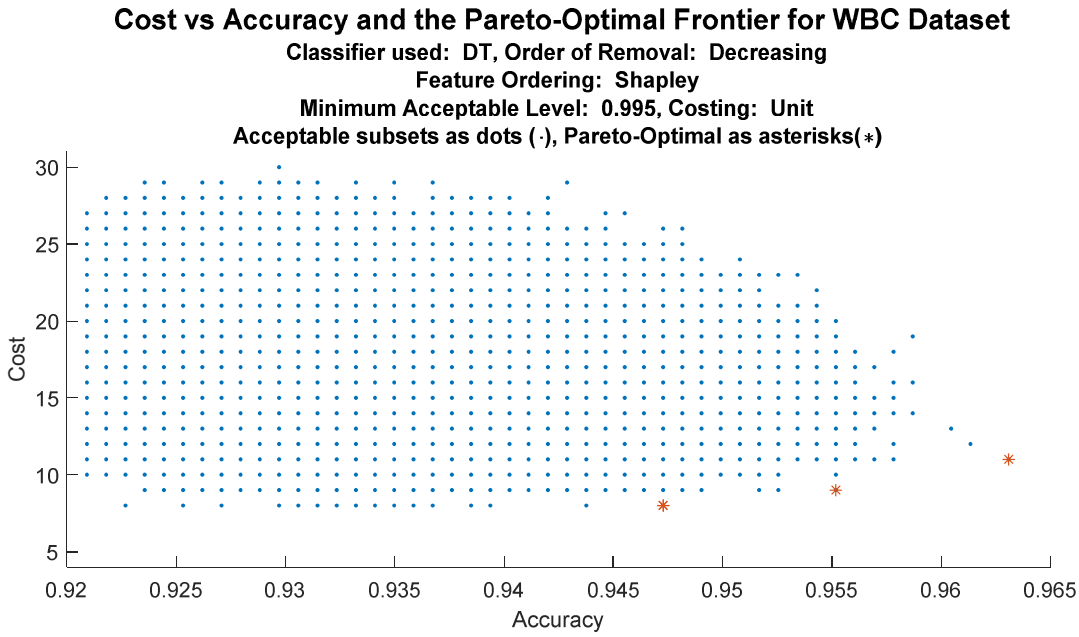

Figure 8. Pareto-Optimal Frontier for the WBC Dataset.

The above figure displays the Pareto-Optimal Frontier for the WBC dataset using the DT classifier, with decreasing order of feature removal, using Shapley feature ordering, with 0.995 as the minimum acceptable level, and with unit costing.

Table 16

Accuracy, Cost and Features Used for POF for the WBC Dataset

Accuracy Value	Cost	Features Used
0.94728	8	00100000000001010100010011010
0.94728	8	00100000000000001011011001010
0.95518	9	00000001000000011001111001010
0.96309	11	00100100100100000100110101010

The above table displays the accuracy, cost and features used for the WBC dataset using the DT classifier, with decreasing order of removal, with Shapley feature ordering, with 0.995 as the minimum acceptable level and with unit costing. The above two examples represent the results from the WBC dataset using DTs with decreasing selection order with both random and unit pricing. The accuracy obtained when all features were used was 0.92455. All of the subsets on the two POFs exceed that value. Some of these superior values were obtained with as few as 8 of the 30 features being present. This reduction represents a very real potential for cost savings while simultaneously providing a superior solution. There were 691134 subsets in the AFS.

For comparison purposes, the above dataset and several others were run with an alternate set of random values. While the results were slightly different, they were consistent with those presented. That is, while different costs were produced and somewhat different sets were on the POF, there were no other remarkable differences in the results. Hence, generating further random pricing sets was not explored further.

However, there is another approach possible regarding pricing. In the table above (Table 16) the first listed feature is a zero for each of the subsets on the POF. It is then possible to assert that, at the pricing given, the first item is too expensive to be included in the POF.

The price of the first feature was lowered to 0.5 and the following results were obtained.

Table 17

POF When Cost of First Feature Decreased to 0.5

Accuracy Value	Cost	Features on POF
0.94376	7.5	1000000000000001110001000 11010
0.94728	8	0010000000001010100010001 1010
0.94728	8	0010000000000001011011000 1010
0.95255	8.5	1010000000000001011011000 1010
0.95518	9	0000000100000001100111100 01010
0.95782	10.5	1010000010011000000111000 11010
0.96309	11	0010010010010000010011010 11010

The costing was as unit costing for all but the first feature which had its value reduced to 0.5. The results can be compared to those of Table 16 where all costs are unit cost. It can be seen that by decreasing the cost of just that one feature, that feature now appeared in three of the seven subsets present in the POF. The cost was further reduced to zero, with the remainder of the costs held at one.

Table 18

POF When Cost of First Feature Decreased to 0

Accuracy Value	Cost	Features on POF
0.94376	7	1000000000000001110001000 11010
0.95255	8	1010000000000001011011000 1010
0.95518	9	0000000100000001100111100 01010
0.95782	10	1010000010011000000111000 11010
0.96309	9	0010010010010000010011010 11010

The costing was as unit costing for all but the first feature which had its value reduced to 0. The results can be compared to those of Table 16 and Table 17. In this case, the feature of interest (the first feature) is now present in thee of the five subsets on the POF. The difference is two of the previously acceptable subsets on the POF (the second and third rows from the previous table with cost of 8) were dominated and so eliminated from the POF. It is apparent that decreasing the cost of a feature has a positive effect on that feature being present in the POF.

Instead of decreasing the cost of an unused feature, it is also possible to increase the cost of a commonly used one. Using the unit cost table, there are 3 features that are present all 4 times: the 22nd, the 27th and the 29th. However, the 27th and the 29th are present in all 691134 sets in the AFS, so regardless of their price, they would be present in every subset on the POF. However, feature 22 is only present in 581624 of those sets. Therefore, it was selected as the feature whose price would be altered. Its cost was set to 10, whereas the rest were left at unit cost.

Table 19

POF When Cost of Specific Feature Increased to 10

Accuracy Value	Cost	Features on POF	Feature 22
0.93849	8	00000000000010000100101010110	0
0.94552	9	00000001010000000110101010101	0
0.94815	10	00100100000010001001101001101	0
0.94815	10	00101000000010001000101010110	0
0.95343	11	00101000001010000011101010101	0
0.95606	12	00100000100000011111100110101	0
0.95694	17	11110010001100001111101010110	0
0.96309	20	00100100100100000100110101101	1

In this case, feature 22 was increased to cost of 10 while the remainder were left at unit cost. From the above table it can be noted that of the 8 subsets in the POF, only 1 uses the 22nd

feature versus all of the POF at unit cost. Hence it is clear that distorting the cost by increasing the price has a negative effect on that feature being present in the POF. In summary, it is apparent that the cost of a feature has a direct impact on the likelihood of that feature being present in the POF.

From the data presented, it can be observed that the number of acceptable sets on the POF is consistently small with the maximum observed being 15. This is so even when the cAFS runs into the hundreds of thousands. Altering the price of a feature was seen to impact the frequency that the given feature appeared in the POF set; increasing the cost decreased the frequency and decreasing the cost increased the frequency.

Answer to Research Question 4: Does the order in which features are removed have an impact on the number of expansions required?

This question is very closely related to the first question. As noted in the answer to RQ 1, in general, removing features in a decreasing order of importance results in a higher Efficiency Ratio than does removing them in random order and removing them in random order results in a higher Efficiency Ratio than does removing them in increasing order. While it is correct to note that this is 'generally' true, there are numerous instances where it is not specifically true. Where the feature ordering was well known and there were no data interactions (specifically, the two synthetic datasets) the results obtained were consistently that decreasing > random > increasing (where > symbolizes a higher Efficiency Ratio, reading from left to right). However, for the natural datasets (IL, Thyroid and WBC), the order was not expressly known and it was not known whether there were data interactions or not. With these datasets it was occasionally found that the ordering was not as expected which might be related to the mal-ordering of the underlying features.

Given an AFS, it is not difficult to determine the frequency of each feature in the AFS. The following table represents the frequency of each feature represented in the AFS for WBC, Decreasing Order, at 0.995 with Shapley ordering.

Table 20

The Relationship Between Feature and Frequency for WBC

feature	1	2	3	4	5	6	7	8
frequency	0.58	0.54	0.75	0.57	0.56	0.59	0.58	0.48
feature	9	10	11	12	13	14	15	16
frequency	0.61	0.56	0.54	0.64	0.57	0.58	0.4	0.59
feature	17	18	19	20	21	22	23	24
frequency	0.63	0.64	0.6	0.6	0.64	0.84	0.62	0.44
feature	25	26	27	28	29	30		
frequency	0.45	0.55	1	0.32	1	0.13		

The above table was created using decreasing order of selection with the threshold set at 0.99. The frequency values range from a low of 0.13 to a high of 1. Three points should be noted. This data gives no information as to the interaction of the various features. Second, while the frequency may be significant at this level, that is no guarantee that the frequencies (and implied significance) will be important at any other level of investigation. Nevertheless, this does suggest an opportunity for future investigation. Third, even if this were to result in a good ordering for the conditions specified, it is of no assistance in determining the values at that level. Specifically, the results need be generated first to determine the required frequencies, hence they cannot be used to determine themselves.

An alternate approach might be to use the information gained at one level to inform the order selection at a lower level. So, if featureX was critical at the 0.997 level it might also be critical at the 0.995 level. Similarly, if it was unimportant and one level, it might also be presumed to be unimportant at a slightly lower level. The following table demonstrates that these assumptions do not appear to hold.

Table 21

Feature Importance for WBC, NNs, Decreasing Order

NNs DEC 0.999			NNs DEC 0.995			NNs DEC 0.99		
Count	Frequency	Feature	Count	Frequency	Feature	Count	Frequency	Feature
47	0.307	F28	86	0.112	F21	21981	0.456	F24
95	0.621	F4	476	0.619	F8	24951	0.518	F4
109	0.712	F13	560	0.728	F4	27676	0.574	F3
112	0.732	F26	572	0.744	F30	28784	0.597	F26
123	0.804	F5	582	0.757	F16	31227	0.648	F9
124	0.81	F20	584	0.759	F13	31606	0.656	F5
124	0.81	F1	596	0.775	F20	31720	0.658	F10
126	0.824	F25	596	0.775	F17	31757	0.659	F30
127	0.83	F10	597	0.776	F3	31861	0.661	F16
129	0.843	F15	598	0.778	F9	32126	0.667	F19
129	0.843	F6	599	0.779	F19	32148	0.667	F17
130	0.85	F9	599	0.779	F5	32379	0.672	F2
131	0.856	F19	604	0.785	F15	32415	0.673	F15
131	0.856	F17	605	0.787	F24	32516	0.675	F13
132	0.863	F12	615	0.8	F18	32527	0.675	F18
132	0.863	F30	620	0.806	F27	32893	0.683	F6
134	0.876	F29	625	0.813	F10	33287	0.691	F12
140	0.915	F3	639	0.831	F12	33952	0.705	F8
140	0.915	F24	648	0.843	F25	34033	0.706	F25
141	0.922	F18	651	0.847	F6	34724	0.721	F20
143	0.935	F16	652	0.848	F29	36029	0.748	F29
143	0.935	F2	657	0.854	F7	36722	0.762	F14
144	0.941	F22	668	0.869	F2	37869	0.786	F7
147	0.961	F27	730	0.949	F14	40381	0.838	F27
151	0.987	F11	740	0.962	F26	41407	0.859	F11

153	1	F7	744	0.967	F11	43534	0.904	F22
153	1	F14	746	0.97	F22	48176	1	F1
153	1	F8	769	1	F1	48176	1	F23
153	1	F23	769	1	F23	48176	1	F21
153	1	F21	769	1	F28	48176	1	F28

It is easy to see that some features are consistent across the range considered. Feature 4 has low frequency for all three levels of significance. Alternately, feature 23 has high frequency at all three levels. However, feature 8 is of critical importance at the 0.999 level, is almost insignificant at 0.995 and somewhat more significant at the 0.99 level. Feature 21 is of utmost significance at the 0.999 level while of negligible significance at the 0.995 level and again of utmost significance at the 0.99 level.

The same issue of inconsistency was observed in other configurations. So, while the notion that some features should be consistently important or unimportant has some intuitive appeal, it does not seem be borne out by this investigation. As the observations related to importance were so inconsistent, no further effort was expended in this area. This suggested approach was called 'Novel Present Count' and was not pursued further.

Answer to Research Question 5: What is the impact of using a different classifier on the AFS produced and the overall efficiency of the process?

For all basic configurations, both classifiers were employed. Some general comments can be made. Altering the process to accommodate the different classifier was not a significant burden. It required configuring the code to run the classifier, setting up the initial conditions, evaluating the results and little more. This author would describe the effort as 'modest'. However, the execution time was significantly greater for the DT classifier. Specifically, the DT classifier discovered vastly more AFSs (and TST and FST) than did the NN one.

Table 22

Comparison of Different Configurations on Resulting AFS and TST

Configuration	NN cAFS	NN cTST	DT cAFS	DT cTST	NN cAFS / NN cTST	DT cAFS / DT cTST	NN cAFS / DT cAFS	NN cTST / DT cTST
Synth Dec Shapley @ .93	32	37	96	100	0.86	0.96	0.333	0.37
Synth Inc Shapley @ 0.93	32	192	150	555	0.17	0.27	0.213	0.346
S 2 Dec Shapley @ .93	32	74	685	730	0.43	0.94	0.047	0.101
S 2 Inc Shapley @ 0.93	78	399	1345	3507	0.2	0.38	0.058	0.114
IL Dec Shapley @ 0.93	343	535	577	673	0.64	0.86	0.594	0.795
IL Inc Shapley @ 0.93	124	180	721	924	0.69	0.78	0.172	0.195
Thyroid Dec Shapley @ 0.995	2275	3469	259147	259309	0.66	1	0.009	0.013
Thyroid Inc Shapley @ 0.995	27547	40198	266069	929140	0.69	0.29	0.104	0.043
WBC Dec Shapley @ 0.995	573	3074	691134	999256	0.19	0.69	0.001	0.003
WBC Inc Shapley @ 0.995	330	1317	135993	250881	0.25	0.54	0.002	0.005
				Average:	0.48	0.67	0.15	0.2

Several features can be noted from Table 22. NNs have a lower Efficiency Ratio than do DTs implying that the DTs are more efficient in finding acceptable feature sets. Further, the average ratio of cAFS for NNs/DTs is consistently less than 1 and sometimes very much so (as little as 0.001 in the selected data from the table), demonstrating that the DTs find many more acceptable feature sets than do the NNs. The higher number of AFS found may relate to the different starting points used. For each classifier and dataset, the best results were estimated as the value using all features. There was frequently a material difference between the two classifiers (see Table 1). As each was scaled the same but from a different starting point, there was no guarantee that the number of AFS would be similar. Of note, finding a greater number of AFS comes at the cost of greater run time. As noted elsewhere, this run time need only be born once.

The count of sets in the POF has been demonstrated to be relatively small (no more than 15 were observed for any configuration) regardless of the number of sets in the AFS (which sometimes exceeded 100,000). Hence, while the DTs find many more sets acceptable, it is not clear that there is a corresponding improvement in the results obtained. Although a point-by-point comparison is always possible, it is not intuitively obvious that any comparison could be generalized. For the configuration of WBC Dec Shapley @ 0.995, the DT produced 1000 times as many sets in the AFS as did the NN. The comparison of the POFs at unit pricing is given below.

Table 23

Comparison of NN and DT Results for Similar Configurations

WBC_Dec_Acceptable_995_NNs_Shapley_unit			WBC_Dec_Acceptable_995_DTs_Shapley_unit		
Features used	Accuracy	Cost	Features used	Accuracy	Cost
00111010111111111110110101110001	0.99294	20	00100000000001010100010001 1010	0.94728	8
10110001101111111011111111 1101	0.99765	23	00100000000000001011011000 1010	0.94728	8
			00000001000000011001111000 1010	0.95518	9
			00100100100100000100110101 1010	0.96309	11

Note: the best value (i.e. that using all features) for WBC for the NN was 0.9858 while that for the DT was 0.9245 and the 0.995 threshold was based against those values respectively. Hence, the comparisons made are relative to the original best values and need to be interpreted carefully. A user desiring to use the NN classifier would expect an accuracy of 0.9858 if all classifiers were used, but 0.99765 if the features in the second row of the POF were used, which needed only 23 of the 30 features. Similarly, a user could employ the DT and anticipate accuracy of 0.9245 if using all features, but 0.96309 if only the 11 features of the fourth row of the POF were used. Note that in both cases, the resulting answers were more accurate and less costly than the 'best' answer available when all features were utilized.

The purpose of using the second classifier was to establish that the general procedure could be extended to another (and, by implication, to *any* other) classifier and that comparable results could be obtained. It was never suggested that the results would be identical in terms of effort required to execute the code, the number of elements in the AFS or the POF or any other detail. Rather, if a user desired to use a specific classifier, it would be possible to do so. In that restricted respect, the concept has been demonstrated.

Summary of Results

The following can be summarized from the results presented. It is efficient to remove features in a decreasing order of importance, however, it is not an easy task to determine that order. Both Simple and Shapley orders were utilized. Other approaches to feature ordering were investigated and found to not offer superior outcomes. The number of reduced feature sets (relative to the number of potential sets) that exceed the thresholds used was consistently small. A higher threshold resulted in a smaller count of AFS. The number of sets on the POF was always a small number. The count was not proportional to the count of AFS, but was consistently no greater than 15, even when the count of AFS was in the hundreds of thousands. The results of using a second classifier demonstrated that, although the detail was different, the general quality of the results was unchanged. With respect to the specific reduced feature sets on the POF were not the same, the count of AFS was not the same and the run times were not the same. However, there was no expectation that these values should be identical. Rather, the overall approach and general nature of the results was comparable. There was a small amount of coding to be accommodated. The initial conditions were different and needed to be accommodated. The run times were different and needed to be accommodated. However, the overall approach was essentially the same. As the process so conveniently generalized to a second classifier it may be assumed to generalize to any other comparable classifier.

Of note, the overall question that drove this investigation was whether or not it was possible to achieve an acceptable answer without using all of the features that might be initially present. For each of the datasets examined, it was possible to achieve results that were nearly as good as the ‘best’ (all features considered) case or materially better than that level using fewer of the features. In some cases, the results obtained were significantly better. In some cases, the

number of features required was significantly fewer than that of the original set. Frequently, both improvements were observed.

Chapter 5

Conclusions, Implications, and Recommendations

From the observations in the previous section, it is possible to come to several conclusions. These are listed as comments in this section and will encompass conclusions, implications and recommendations. A brief summary of this chapter will follow. Finally, a short summary of the entire investigation will be provided.

Comment on ordering of features

It is clear that, if the order of importance of the features is known, then the most efficient order of removal is decreasing followed by random followed by increasing. The order of importance is known for Synth and S 2 and they demonstrate that selection in decreasing order is most efficient when the ordering of features is categorically known. For the other datasets, the ordering is not categorically known but rather estimated (using both the Simple and Shapley process). While these other datasets generally demonstrate decreasing selection order as being the most efficient there are specific examples, noted elsewhere, where such is not the case. So, for the purpose at hand, the current methods (Simple and Shapely) do not provide a completely satisfactory answer to feature ordering. The implications are that, if decreasing order is used with

Simple or Shapley ordering, some sets that would be acceptable might be missed. Alternately, if increasing order is used then additional processing time will be required.

There are several approaches that might be taken to address this issue. First, the situation might simply be endured without a specific remedy. Decreasing order using Simple or Shapley produces a workable set of results and increasing order produces a larger set, albeit consuming more resources. However, the resources consumed are not especially expensive and only need be consumed once. This is the 'do nothing' solution.

Second, assume that the acceptable level might be approached in several steps (as was done in this investigation). The exact number of steps and their values are not critical to this discussion, but let us assume 99%, 97% and 95% for purposes of discussion, where the percentage is relative to the maximum that would be obtained using all of the features. At each level, the frequency of each feature present in the AFS can easily be counted. Frequency can be used as a proxy for importance and the features can then be ranked on that basis. That ordering, termed 'Novel Present Count', can then be used for the next iteration, and the overall process repeated. This idea was tested and found to be deficient. The significance of a feature at one level was frequently materially different from its significance at another level. Hence, this idea, as currently developed, is not useful.

Third, it can be noted that the acceptable level presents a sharp demarcation between acceptable and not acceptable. So for some acceptable level, AL, and some subset, SS, if the result is AL +0.001 then SS is acceptable and its subsets are examined while if the result is AL – 0.001 then SS is rejected and none of its subsets are examined. However, there is no practical difference between AL + 0.001 and AL – 0.001, yet one is acceptable, and its subsets are

candidates for acceptance while the other, along with its subsets, is rejected. There are at least two potential ways to soften this hard threshold. Both are relatively simple but have not been tested. The first would to be to fail a subset only if both it and its parent were below the acceptable level. This softer fail would require some additional accounting in the process and would certainly increase the number of subsets to be evaluated. The second would set a value materially below AL as the temporary threshold value. Then a set of temporarily acceptable sets would be generated. Then that set could be filtered based on the actual AL. In this way, some subsets might be captured that would otherwise have been missed. Two items should be noted. First, neither of these approaches has been tested; they are simply suggestions for potential future work. Second, it is not clear that either would improve efficiency, indeed, either may degrade the overall efficiency. However, both should increase (or, at least, not decrease) the number of sets in the AFS.

Comment on the number within the AFS on the Pareto-Optimal Frontier

The number of subsets on the POF was consistently very small, with 15 being the observed maximum, even when the count of AFS exceeded 100,000. Of note, a costing function must be available to generate the POF. Both random and unit pricing were used and the different pricing resulted in different subsets on the POF. Increasing the price of a feature caused that feature to appear less often in the POF group. Alternately, decreasing the price of a feature resulted in that feature being more frequently in the POF group. Hence, the subsets in the POF were responsive to pricing. Of note, some features were so dominant that they were present in all subsets in the AFS. Obviously, the price of such features would have no impact on the subsets in the POF.

Comment on using a second classifier

The tests were initially carried out using Neural Networks as the classifier. Then Decision Trees were used as a second classifier. There was very little effort required to replace one classifier with the other, although the number of executions was necessarily increased. It was observed that the count of the AFS was significantly higher using the DT relative to the NN, frequently one or two orders of magnitude. One explanation for this may be that the DT typically had a lower starting accuracy. Then both acceptable levels were scaled using the same ratios for acceptability.

As using the second classifier presented no challenges, it can be assumed that any comparable classifier could also be used. Of note, Matlab was the environment used and both classifiers were from that environment. Attempting to port in a classifier from a different environment was not attempted, so no comment can be made as to the feasibility of doing so. Matlab was selected as it was a convenient environment that natively supported the desired classifiers. There is no reason to believe that other environments might not have been useful.

Comment on the subsets on the POF, primary conclusion

The subsets on the POF demonstrated the value of this approach. It was frequently the case that they provided a slightly better accuracy than did the complete feature set. However, it was always the case that they produced answers using significantly fewer features. The original impetus for this effort was that it was suspected that not all features were required in order to achieve an acceptable diagnosis. This has been borne out by experimentation.

Further, it was observed that, as the level of accuracy was decreased, there was frequently a remarkable reduction in the number of features required (the only exceptions to this occurred

when there was only one subset on the POF). The reduction in the number of features required for an acceptable level of accuracy represents the opportunity for material cost savings when performing the desired diagnosis. As the results are presented on a POF, the clinician has an optimal set of choices. The highest level of accuracy can be selected, which is frequently higher than that obtained by using all of the features, and this level of accuracy is always at a lower cost. Alternately, the clinician can make a selection using either price, accuracy, or both as the criteria. In every case, because only the POF subsets need be considered and the number of choices is not overwhelming (15 was the maximum observed in this study) the effort on the part of the clinician would be modest. In several cases, relatively good accuracy was obtained, relative to the 'best' accuracy, with only a very few of the features being required. As example, with WBC using NNs, an accuracy of 0.99294 was obtained using just 20 of the original 30 features while the accuracy using all 30 features was slightly worse at 0.9858, while an accuracy of 0.99765 was obtained using just 23 features.

Comment on generalizability of results

This effort was motivated by a desire to determine if it might be possible to decrease the cost of testing when applied to medical diagnoses. In the course of this effort, two synthetic and three natural sets of data were used. Although three of the datasets were specific to medical diagnoses, there was nothing about the structure of the datasets that could not be generalized to any comparable classification. That is, there is no obvious reason why this process could not be applied to any other situation where a classifier might be used. As there are many other areas where classifiers are used, this generalizability represents an opportunity for material cost savings and, if the existing datasets are representative, the possibility of simultaneously improving the accuracy.

Brief Summary of Conclusions, Implications, Recommendations

From the material presented, the conclusions, implications and recommendations can be summarized as follows. Determining the importance of individual features in a feature set is a challenging problem that is not yet solved in the general case. The frequency of features in an acceptable feature set was found to not be an acceptable approach to determining relative importance. Removing features in decreasing order of importance is more efficient than either random or increasing order. The POF is responsive to the cost of individual features. The process described is not tightly tied to any classifier. The process described is not tightly tied to the area of medical diagnosis. Most of the effort required for this process is one-time and up-front, so a user of the system need do little more than provide a cost profile and will then be provided with the POF. The process described was always able to produce an answer comparable to the best available, and usually better, while always reducing the number of features required and, hence, the cost. Determining this last item was the primary goal of this effort and is the most significant conclusion.

Overall Summary

Introduction

Classifiers have a lengthy history of being used to assist with medical diagnosis (Er, Yumusak, and Temurtas, 2010), (Syiam, 1994), (Gutierrez, 2015), (Kadi & Idri, 2015), (Kumar & Krishniah, 2016). Significant effort has been extended in searching for a single, ideal, subset of features. The current understanding of the single subset problem was reviewed by Li et al. (2017). There has been little, if any, effort spent looking for all subsets of features that provide a suitable answer. While 'suitable' may be difficult to define precisely, an operational definitional might be said to mean that the answer is good enough for the purpose at hand, that is, producing an accuracy above some user-defined threshold. Let this subset of sets be termed the Acceptable Feature Set (AFS). Assuming that such an AFS can be determined, if a costing function exists then the cost of determining each subset can be calculated. An accuracy versus cost curve can then be generated. From that curve, the Pareto-Optimal Frontier (POF) can be established. An end user could then make an informed decision as to which subset was best under the prevailing circumstances. The objective of this investigation has been to find the POF which then constitutes the set of subsets that cost the least and provide the most accurate answers.

In order to discover the POF, five major questions were examined. Specifically:

- What is an efficient process to identify all acceptable feature sets?
- What percentage of the reduced feature sets are above the minimum quality threshold established?
- What percentage of the qualifying feature sets are on the Pareto-Optimal Frontier?

- Does the order in which features are removed have an impact on the number of expansions required?
- What is the impact of using a different classifier on the AFS produced and the overall efficiency of the process?

Fundamental Assumptions

There were several assumptions that were fundamental to this effort.

- *Assumption*: If a classifier $D^{F_s,m}$ is not acceptable, then all classifiers trained on a proper subset of F_s are also not acceptable.
- *Assumption*: There is a lower bound with respect to the accuracy below which the solution will not be of interest. This bound, l_j, will be arbitrarily established. It establishes a threshold level above which the estimated accuracy is acceptable while below which the estimated accuracy is not acceptable.
- *Assumption*: A 'nearly best' answer will be obtained by using all of the features. The results might improve slightly as some noisy features are removed. However, the assumption is that accuracy will soon start to fall off as more meaningful features are removed.
- *Assumption*: Features can be ranked in a meaningful order (least to most significant or the reverse). Any such ranking is subjective as the methodology used to establish significance is a user choice (but once the choice is made there would be no further subjectivity).

Data Sets Used

A total of five datasets were used. There were two synthetic datasets that the author generated primarily to establish that the process was correct, although they also provided

some additional information. A third dataset was sourced from the UCI catalog related to Indian Liver Patient Disease (IL). The ILPD consists of 10 attributes and 583 data rows. It is sufficiently small to establish that the overall approach is sound. It is available at http://archive.ics.uci.edu/ml/datasets/ILPD+%28Indian+Liver+Patient+Dataset%29. Also from the UCI resource, data was used with respect to the Wisconsin Breast (Diagnostic) Cancer (WBC). The Breast Cancer Wisconsin (Diagnostic) Data Set is available at http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29. It consists of 30 attributes and has 569 data rows. It is sufficiently large to establish that the process will scale. MatLab has many built in datasets. Their thyroid dataset was also used, primarily as confirmation of the some of the other observations. The thyroid dataset had 21 features and 498 rows of data were used.

Classifiers Used

The two classifiers used were Neural Networks (NN) and Decision Trees (DT). Both were used in the MatLab (R2017A) environment. These two were selected because they both have an extensive history of being used in classification in general and medical diagnosis in particular including, for NNs:

- ovarian cancer (Tan, Quek, & Ng, 2005),
- cirrhosis (Sun, Lu, Kobayashi, & Yahagi, 2005),
- carpal tunnel syndrome (Palfy & Papez, 2007), and
- sleep apnea (Marcos, Hornero, Alvarez, Campo, & Lopez, 2007),

and for DTs:

- anemia (Maity, Sarkar & Chakraborty, 2012),

- bladder cancer (Floares & Birlutiu, 2012),
- monitoring posture and activities (Zhang & Sazonov, 2012), and
- pulmonary disorders (Tartar, Kilic & Akan, 2013).

The primary classifier was NN. The secondary classifier (DT) was included to determine whether or not the process was tightly tied to the NN.

Methodology

The steps required to perform the analysis are summarized in this section. A depth-first search was used as a breadth-first search would have exhausted machine memory. A 'nearly best' accuracy was determined by using all features with the NN. The features were ordered in a least-to-most significant ordering. Two approaches to ordering were used. The Simple ordering process involved two passes. The first pass removed a given feature from the feature set and the results were determined. The second pass used only the given feature and the results again determined. The two results were then combined and the process repeated for the remainder of the feature set. The second approach used the well-known Shapley ordering process.

Based on the 'nearly best' accuracy, a series of thresholds were established. For the Synth, S 2 and IL datasets the relative values were 0.99, 0.96 and 0.93 of the nearly best accuracy. For Thyroid and WBC the relative values were 0.999, 0.997 and 0.995. Testing always started with the complete set of features (the complete subset). If the subset under examination was above the threshold, it was added to the 'acceptable' list and its subsets were added to the 'to be evaluated' list. The features extant in the subset were eliminated in particular order. The subsets were added on the basis of either 'increasing', 'decreasing' or 'random' order. Suppose the order was 'decreasing'. Consider featureX, extant in the subset and next to be considered for elimination. If

there were features less significant than featureX, then they would be eliminated in turn and their subsets added to the 'to be evaluated' queue. If there were none (that were less significant than featureX), then no further processing at that node was required. Generation/elimination for 'increasing' was performed in the opposite order. For 'random' processing, the process was as if for increasing but the order of the features was randomized. Such processing generated a set of feature subsets and accuracy pairs. The effort was repeated for the second classifier.

After the execution had completed, two costing functions were applied to the set of acceptable subsets. The costing functions used were unit and random. The random values were determined once only, they were not random for each subset. The result was an accuracy versus cost curve for each costing function. From this relationship, it was a trivial matter to extract the POF.

Results

A complete listing of the results is out of scope for this brief summary. However, the basic five questions can be addressed. Ordering of the features remains a challenge. When the ordering is known, removing the features in decreasing order is the most efficient. Decreasing order of removal also is most efficient when the ordering is approximately correct but may miss some acceptable subsets. A relatively small number of subsets are acceptable (that is, relative to the count of potential subsets) but the actual fraction obviously depends on the threshold limit. The lower the limit, the greater is the percentage. The number of qualifying subsets on the Pareto-Optimal Frontier is always a small count; it never exceeded 15 even when the number of acceptable subsets exceeded 100,000. Both classifiers provided comparable (although not identical) answers.

Conclusions

The most salient conclusion is that the Pareto-Optimal Frontier offers an accuracy level that frequently exceeds the 'nearly best' values and always does so at a reduced cost. In many cases, the number of features required for an acceptable answer was drastically reduced. As example, for the WBC dataset, only 23 of the 30 features were required to achieve an accuracy of 0.99765 using NNs, while the 'best accuracy' using all features was only 0.9858. Therefore the answer was cheaper and the results were better. Using the DT classifier on the same dataset, the 'best accuracy' using all features was 0.9245. However, just 11 of the features could be used to observe an accuracy of 0.96309. Again, better accuracy at an obviously lower cost was observed. Similar observations were made for all datasets. Hence, the underlying motivation for the investigation was validated: not all features are required. A pleasant surprise was that it was sometimes possible to obtain both a lower cost and a higher level of accuracy.

The method of determining the acceptable subset is dependent on the ability to order the features. The successful ordering has not been clearly established. When the ordering was known (as in the two synthetic cases) the decreasing order of removal was clearly the best. When the ordering was estimated (for the natural datasets, whether using Simple or Shapley ordering), decreasing order typically performed the best, but there were instances where it did not. This suggests, but is not conclusive, that the ordering was not correct.

The process is not tightly tied to the classifier. Replacing the NN with the DT required only a modest effort. The results obtained were not identical, but were generally comparable with those obtained from the NN.

Appendix A

Summary Data

The following tables represent the summarized data from the various executions. They are not required for the general arguments and conclusions in this paper, but are included so that the interested reader can examine the detailed results, if desired. These are organized as follows. The five different datasets are represented. Each one was tested at three levels of acceptability. Each one was tested with NNs and DTs. Each one was tested using Simple and Shapley sorting. Note, the 'Random' testing was done with Simple sorting and so not repeated with Shapley as randomizing the feature set means that the starting point (Simple or Shapley) is immaterial. The results are presented in tabular format. The abbreviations used follow those in the main text.

Table 24

Synth Using Simple Ordering

ORDERING SIMPLE	cAFS	cTST	cFST	Ratio cAFS / cTST	cPOF @ unit cost	Ratio cPOF / cAFS @ unit cost	cPOF @ random cost	Ratio cPOF / cAFS @ random cost
Synth INC @ 0.93 NNs	32	192	160	0.167	1	0.031	1	0.031
Synth INC @ 0.96 NNs	32	192	160	0.167	1	0.031	1	0.031
Synth INC @ 0.99 NNs	32	192	160	0.167	1	0.031	1	0.031
Synth DEC @ 0.93 NNs	32	37	5	0.865	1	0.031	1	0.031
Synth DEC @ 0.96 NNs	32	37	5	0.865	1	0.031	1	0.031
Synth DEC @ 0.99 NNs	32	37	5	0.865	1	0.031	1	0.031
Synth RAN @ 0.93 NNs	34	57	23	0.596	1	0.029	1	0.029
Synth RAN @ 0.96 NNs	32	50	18	0.640	1	0.031	1	0.031
Synth RAN @ 0.99 NNs	28	47	19	0.596	2	0.071	2	0.071
Synth INC @ 0.93 DTs	214	680	466	0.315	5	0.023	6	0.028
Synth INC @ 0.96 DTs	141	509	368	0.277	4	0.028	6	0.043
Synth INC @ 0.99 DTs	31	154	123	0.201	1	0.032	2	0.065
Synth DEC @ 0.93 DTs	164	171	7	0.959	2	0.012	3	0.018
Synth DEC @ 0.96 DTs	88	93	5	0.946	4	0.045	5	0.057
Synth DEC @ 0.99 DTs	21	31	10	0.677	2	0.095	4	0.190
Synth RAN @ 0.93 DTs	220	321	101	0.685	4	0.018	5	0.023
Synth RAN @ 0.96 DTs	129	231	102	0.558	3	0.023	4	0.031
Synth RAN @ 0.99 DTs	60	91	31	0.659	2	0.033	2	0.033

Using the Ratio of cAFS/cTST as the measure, it can be observed that Decreasing order of removal is consistently superior to Random which is consistently superior to Increasing. The number of subsets on the POF is consistently small.

Table 25

Synth Using Shapley Ordering

ORDERING SHAPLEY	cAFS	cTST	cFST	Ratio cAFS / cTST	cPOF @ unit cost	Ratio cPOF / cAFS @ unit cost	cPOF @ random cost	Ratio cPOF / cAFS @ random cost
Synth INC @ 0.93 NNs	32	192	160	0.167	6	0.188	2	0.063
Synth INC @ 0.96 NNs	32	192	160	0.167	1	0.031	1	0.031
Synth INC @ 0.99 NNs	32	192	160	0.167	1	0.031	1	0.031
Synth DEC @ 0.93 NNs	32	37	5	0.865	1	0.031	1	0.031
Synth DEC @ 0.96 NNs	32	37	5	0.865	1	0.031	1	0.031
Synth DEC @ 0.99 NNs	31	37	6	0.838	1	0.032	1	0.032
Synth INC @ 0.93 DTs	150	555	405	0.270	2	0.013	2	0.013
Synth INC @ 0.96 DTs	98	417	319	0.235	3	0.031	3	0.031
Synth INC @ 0.99 DTs	1	11	10	0.091	1	1.000	1	1.000
Synth DEC @ 0.93 DTs	96	100	4	0.960	1	0.010	2	0.021
Synth DEC @ 0.96 DTs	82	90	8	0.911	1	0.012	1	0.012
Synth DEC @ 0.99 DTs	1	11	10	0.091	1	1.000	1	1.000

Again, the Ratio of cAFS/cTST consistently demonstrates that Decreasing is more efficient than Increasing (with Random not being repeated). The cPOF consistently remains small.

Table 26

S 2 Using Simple Ordering

Ordering Simple	cAFS	cTST	cFST	Ratio cAFS / cTST	cPOF @ unit cost	Ratio cPOF / cAFS @ unit cost	cPOF @ random cost	Ratio cPOF / cAFS @ random cost
S 2 INC @ 0.93 NNs	73	312	239	0.234	3	0.041	5	0.068
S 2 INC @ 0.96 NNs	39	291	253	0.134	3	0.077	5	0.128
S 2 INC @ 0.99 NNs	31	280	249	0.111	3	0.097	5	0.161
S 2 DEC @ 0.93 NNs	51	84	33	0.607	2	0.039	3	0.059
S 2 DEC @ 0.96 NNs	31	40	9	0.775	3	0.097	4	0.129
S 2 DEC @ 0.99 NNs	8	18	10	0.444	3	0.375	3	0.375
S 2 RAN @ 0.93 NNs	59	175	127	0.337	2	0.034	4	0.068
S 2 RAN @ 0.96 NNs	31	127	96	0.244	2	0.065	3	0.097
S 2 RAN @ 0.99 NNs	14	59	45	0.237	1	0.071	1	0.071
S 2 INC @ 0.93 DTs	1303	2176	873	0.599	6	0.005	14	0.011
S 2 INC @ 0.96 DTs	541	966	425	0.560	3	0.006	7	0.013
S 2 INC @ 0.99 DTs	112	296	184	0.378	4	0.036	7	0.063
S 2 DEC @ 0.93 DTs	222	250	28	0.888	4	0.018	8	0.036
S 2 DEC @ 0.96 DTs	90	100	10	0.900	2	0.022	3	0.033
S 2 DEC @ 0.99 DTs	29	36	7	0.806	3	0.103	4	0.138
S 2 RAN @ 0.93 DTs	1165	2133	968	0.546	5	0.004	9	0.008
S 2 RAN @ 0.96 DTs	459	956	497	0.480	5	0.011	5	0.011
S 2 RAN @ 0.99 DTs	86	265	179	0.325	2	0.023	2	0.023

As measured by the ratio of cAFS to cTST, the Decreasing order of removal consistently outperforms the Random order which typically outperforms the Increasing order of removal. The cPOF remains relatively small. Note that the cPOF is typically higher for the random cost than the unit cost. This is not surprising as the random cost can more finely divide the space than can the unit cost.

Table 27

S 2 Using Shapley Ordering

Ordering Shapley	cAFS	cTST	cFST	Ratio cAFS / cTST	cPOF @ unit cost	Ratio cPOF / cAFS @ unit cost	cPOF @ random cost	Ratio cPOF / cAFS @ random cost
S2 INC @0.93 NNs	78	399	321	0.195	7	0.090	7	0.090
S2 INC @0.96 NNs	39	298	259	0.131	4	0.103	4	0.103
S2 INC @0.99 NNs	27	246	219	0.110	3	0.111	2	0.074
S2 DEC @0.93 NNs	45	73	28	0.616	2	0.044	2	0.044
S2 DEC @0.96 NNs	36	70	34	0.514	2	0.056	2	0.056
S2 DEC @0.99 NNs	20	32	12	0.625	2	0.100	2	0.100
S2 INC @0.93 DTs	1345	3507	2162	0.384	6	0.004	12	0.009
S2 INC @0.96 DTs	573	2112	1539	0.271	4	0.007	8	0.014
S2 INC @0.99 DTs	59	337	278	0.175	3	0.051	4	0.068
S2 DEC @0.93 DTs	684	729	45	0.938	4	0.006	6	0.009
S2 DEC @0.96 DTs	142	171	29	0.830	2	0.014	2	0.014
S2 DEC @0.99 DTs	45	70	25	0.643	2	0.044	4	0.089

Regardless of whether NNs or DTs were used, the ratio of cAFS/cTST was higher with Decreasing order of removal as compared with Increasing order of removal. The cPOF is typically higher with random cost and is consistently small.

Table 28

IL Using Simple Ordering

Ordering Simple	Iter	cAFS	cTST	cFST	Ratio cAFS / cTST	cPOF @ unit cost	Ratio cPOF / cAFS @ unit cost	cPOF @ random cost	Ratio cPOF /cAFS @ random cost
IL INC @ 0.93 NNs	1	478	648	169	0.738	3	0.006	9	0.019
IL INC @ 0.93 NNs	2	227	294	67	0.772	3	0.013	5	0.022
IL INC @ 0.93 NNs	3	600	795	195	0.755	3	0.005	6	0.010
IL INC @ 0.96 NNs	1	204	393	189	0.519	6	0.029	10	0.049
IL INC @ 0.96 NNs	2	82	155	73	0.529	4	0.049	8	0.098
IL INC @ 0.96 NNs	3	51	108	57	0.472	4	0.078	4	0.078
IL INC @ 0.99 NNs	1	13	47	34	0.277	2	0.154	2	0.154
IL INC @ 0.99 NNs	2	3	19	16	0.158	1	0.333	1	0.333
IL INC @ 0.99 NNs	3	4	22	18	0.182	4	1.000	3	0.750
IL DEC @ 0.93 NNs	1	247	301	54	0.821	5	0.020	9	0.036
IL DEC @ 0.93 NNs	2	320	393	73	0.814	3	0.009	5	0.016
IL DEC @ 0.93 NNs	3	388	492	104	0.789	3	0.008	4	0.010
IL DEC @ 0.96 NNs	1	20	39	19	0.513	3	0.150	5	0.250
IL DEC @ 0.96 NNs	2	34	78	44	0.436	3	0.088	5	0.147
IL DEC @ 0.96 NNs	3	47	75	28	0.627	3	0.064	6	0.128
IL DEC @ 0.99 NNs	1	2	15	13	0.133	1	0.500	1	0.500
IL DEC @ 0.99 NNs	2	4	18	14	0.222	3	0.750	4	1.000
IL DEC @ 0.99 NNs	3	3	22	19	0.136	2	0.667	2	0.667
IL RAN @ 0.93 NNs	1	498	668	170	0.746	6	0.012	10	0.020
IL RAN @ 0.93 NNs	2	486	608	122	0.799	4	0.008	7	0.014
IL RAN @ 0.93 NNs	3	485	643	158	0.754	3	0.006	9	0.019
IL RAN @ 0.96 NNs	1	26	52	26	0.500	2	0.077	4	0.154
IL RAN @ 0.96 NNs	2	29	59	30	0.492	4	0.138	8	0.276
IL RAN @ 0.96 NNs	3	70	122	52	0.574	3	0.043	4	0.057
IL RAN @ 0.99 NNs	1	5	27	27	0.185	2	0.400	2	0.400
IL RAN @ 0.99 NNs	2	1	11	10	0.091	1	1.000	1	1.000
IL RAN @ 0.99 NNs	3	5	28	23	0.179	2	0.400	2	0.400
IL INC @0.93 DTs	1	650	770	120	0.844	4	0.006	8	0.012
IL INC @0.93 DTs	2	670	802	132	0.835	5	0.007	7	0.010
IL INC @0.93 DTs	3	586	699	113	0.838	5	0.009	11	0.019
IL INC @0.96 DTs	1	152	244	92	0.623	2	0.013	5	0.033
IL INC @0.96 DTs	2	184	277	93	0.664	3	0.016	6	0.033
IL INC @0.96 DTs	3	303	485	182	0.625	3	0.010	6	0.020
IL INC @0.99 DTs	1	30	85	55	0.353	3	0.100	4	0.133
IL INC @0.99 DTs	2	20	60	40	0.333	2	0.100	2	0.100
IL INC @0.99 DTs	3	47	105	58	0.448	5	0.106	4	0.085

IL DEC @0.93 DTs	1	585	682	98	0.858	2	0.003	6	0.010
IL DEC @0.93 DTs	2	674	769	95	0.876	3	0.004	5	0.007
IL DEC @0.93 DTs	3	598	707	109	0.846	5	0.008	6	0.010
IL DEC @0.96 DTs	1	284	400	116	0.710	2	0.007	4	0.014
IL DEC @0.96 DTs	2	279	377	98	0.740	1	0.004	3	0.011
IL DEC @0.96 DTs	3	265	382	117	0.694	5	0.019	7	0.026
IL DEC @0.99 DTs	1	43	76	33	0.566	2	0.047	4	0.093
IL DEC @0.99 DTs	2	25	50	25	0.500	2	0.080	3	0.120
IL DEC @0.99 DTs	3	38	88	50	0.432	2	0.053	4	0.105
IL RAN @0.93 DTs	1	585	727	142	0.805	5	0.009	8	0.014
IL RAN @0.93 DTs	2	658	842	184	0.781	6	0.009	7	0.011
IL RAN @0.93 DTs	3	637	803	166	0.793	3	0.005	7	0.011
IL RAN @0.96 DTs	1	270	406	136	0.665	3	0.011	5	0.019
IL RAN @0.96 DTs	2	210	333	123	0.631	4	0.019	6	0.029
IL RAN @0.96 DTs	3	80	97	17	0.825	3	0.038	5	0.063
IL RAN @0.99 DTs	1	28	84	56	0.333	2	0.071	2	0.071
IL RAN @0.99 DTs	2	10	35	25	0.286	2	0.200	4	0.400
IL RAN @0.99 DTs	3	39	92	53	0.424	1	0.026	1	0.026

The IL with Simple ordering was expanded from the other configurations. In this case, each configuration was executed three times in order to gain insight into the consistency of the answers produced. The ‘iter’ column refers to the specific iteration. It is not difficult to observe that there can be a significant amount of variability from one iteration to the next. Yet, the previous patterns of Decreasing being more efficient than Random which is itself more efficient than Increasing remains consistent.

Table 29

IL Using Shapley Ordering

Ordering Shapley	cAFS	cTST	cFST	Ratio cAFS / cTST	cPOF @ unit cost	Ratio cPOF / cAFS @ unit cost	cPOF @ random cost	Ratio cPOF / cAFS @ random cost
IL INC @ 0.93 NNs	124	180	56	0.689	4	0.032	5	0.040
IL INC @ 0.96 NNs	18	51	33	0.353	2	0.111	2	0.111
IL INC @ 0.99 NNs	5	21	16	0.238	2	0.400	2	0.400
IL DEC @ 0.93 NNs	343	535	192	0.641	5	0.015	10	0.029
IL DEC @ 0.96 NNs	30	82	52	0.366	2	0.067	2	0.067
IL DEC @ 0.99 NNs	4	19	15	0.211	3	0.750	2	0.500
IL INC @ 0.93 DTs	721	924	203	0.780	3	0.004	5	0.007
IL INC @ 0.96 DTs	299	549	250	0.545	4	0.013	10	0.033
IL INC @ 0.99 DTs	18	57	39	0.316	2	0.111	3	0.167
IL DEC @ 0.93 DTs	577	672	95	0.859	4	0.007	5	0.009
IL DEC @ 0.96 DTs	208	276	78	0.754	5	0.024	8	0.038
IL DEC @ 0.99 DTs	17	34	17	0.500	3	0.176	3	0.176

With respect to the DTs, the Decreasing order of removal is consistently superior (as measured by the cAFS/cTST ratio). This is not consistently observed with the NNs (although the values are relatively close to one another).

Table 30

Thyroid Using Simple Ordering

Ordering Simple	cAFS	cTST	cFST	Ratio cAFS / cTST	cPOF @ unit cost	Ratio cPOF / cAFS @ unit cost	cPOF @ random cost	Ratio cPOF / cAFS @ random cost
Thyroid INC @ 0.995 NNs	11632	13927	2295	0.835	3	0.00026	8	0.00069
Thyroid INC @ 0.997 NNs	2478	3179	701	0.779	4	0.00161	9	0.00363
Thyroid INC @ 0.999 NNs	7	53	46	0.132	3	0.42857	2	0.28571
Thyroid DEC @ 0.995 NNs	8917	10790	1873	0.826	2	0.00022	6	0.00067
Thyroid DEC @ 0.997 NNs	1162	2062	901	0.564	5	0.00430	5	0.00430
Thyroid DEC @ 0.999 NNs	1724	2116	392	0.815	3	0.00174	5	0.00290
Thyroid RAN @ 0.995 NNs	14403	17829	3427	0.808	4	0.00028	5	0.00035
Thyroid RAN @ 0.997 NNs	2961	3757	796	0.788	4	0.00135	4	0.00135
Thyroid RAN @ 0.999 NNs	1311	1852	541	0.708	5	0.00381	8	0.00610
Thyroid INC @ 0.995 DTs	262695	822580	559885	0.319	2	0.00001	4	0.00002
Thyroid INC @ 0.997 DTs	231355	727507	496152	0.318	3	0.00001	4	0.00002
Thyroid INC @ 0.999 DTs	31259	104030	72771	0.300	1	0.00003	1	0.00003
Thyroid DEC @ 0.995 DTs	260980	261145	165	0.999	4	0.00002	2	0.00001
Thyroid DEC @ 0.997 DTs	224462	227912	3450	0.985	13	0.00006	2	0.00001
Thyroid DEC @ 0.999 DTs	27824	31882	4058	0.873	3	0.00011	3	0.00011
Thyroid RAN @ 0.995 DTs	261518	392587	131068	0.666	4	0.00002	2	0.00001
Thyroid RAN @ 0.997 DTs	231916	352441	121525	0.658	7	0.00003	2	0.00001
Thyroid RAN @ 0.999 DTs	24037	40637	16600	0.592	1	0.00004	3	0.00012

With increasing numbers of features, the counts of the various subsets (acceptable, tested, and failed) increased dramatically. With respect to the DTs, the efficiency order of Decreasing being better than Random being better than Increasing is apparent. With respect to NNs, it is not. Note that the cPOF whether for unit or random pricing remains very small.

Table 31

Thyroid Using Shapley Ordering

Ordering Shapley	cAFS	cTST	cFST	Ratio cAFS / cTST	cPOF @ unit cost	Ratio cPOF / cAFS @ unit cost	cPOF @ random cost	Ratio cPOF / cAFS @ random cost
Thyroid INC @ 0.995 NNs	27547	40197	12650	0.685	3	0.000109	6	0.000218
Thyroid INC @ 0.997 NNs	8126	12028	3902	0.676	2	0.000246	6	0.000738
Thyroid INC @ 0.999 NNs	9981	16277	6296	0.613	5	0.000501	5	0.000501
Thyroid DEC @ 0.995 NNs	2275	3469	1194	0.656	4	0.001758	4	0.001758
Thyroid DEC @ 0.997 NNs	1125	1984	859	0.567	4	0.003556	7	0.006222
Thyroid DEC @ 0.999 NNs	805	1339	534	0.601	6	0.007453	6	0.007453
Thyroid INC @ 0.995 DTs	266070	929141	663071	0.286	5	0.000019	2	0.000008
Thyroid INC @ 0.997 DTs	247268	871084	623816	0.284	1	0.000004	1	0.000004
Thyroid INC @ 0.999 DTs	37262	139292	102030	0.268	1	0.000027	3	0.000081
Thyroid DEC @ 0.995 DTs	259147	259308	161	0.999	3	0.000012	11	0.000042
Thyroid DEC @ 0.997 DTs	205213	208132	2919	0.986	2	0.000010	1	0.000005
Thyroid DEC @ 0.999 DTs	43229	49989	6760	0.865	4	0.000093	2	0.000046

Comparing the Shapley ordering to the Simple ordering, it is again observed that the NNs do not exhibit the otherwise commonly observed superiority of Decreasing order compared to

Increasing order of removal. However, as with the Simple ordering, the DTs strongly exhibit that characteristic.

Table 32

WBC Using Simple Ordering

Ordering Simple	cAFS	cTST	cFST	Ratio cAFS / cTST	cPOF @ unit cost	Ratio cPOF / cAFS @ unit cost	cPOF @ random cost	Ratio cPOF / cAFS @ random cost
WBC INC @ 0.995 NNs	2848	9776	6928	0.291	4	0.00140	3	0.00105
WBC INC @ 0.997 NNs	298	1513	1215	0.197	4	0.01342	8	0.02685
WBC INC @ 0.999 NNs	73	473	400	0.154	3	0.04110	4	0.05479
WBC DEC @ 0.995 NNs	769	3081	2312	0.250	5	0.00650	6	0.00780
WBC DEC @ 0.997 NNs	139	793	655	0.175	6	0.04317	4	0.02878
WBC DEC @ 0.999 NNs	153	935	782	0.164	6	0.03922	6	0.03922
WBC RAN @ 0.995 NNs	1360	4407	3047	0.309	8	0.00588	7	0.00515
WBC RAN @ 0.997 NNs	194	884	690	0.219	5	0.02577	4	0.02062
WBC RAN @ 0.999 NNs	96	636	541	0.151	3	0.03125	5	0.05208
WBC INC @ 0.995 DTs	100993	183093	82100	0.552	7	0.00007	10	0.00010
WBC INC @ 0.997 DTs	41023	70753	29730	0.580	6	0.00015	6	0.00015
WBC INC @ 0.999 DTs	927	2054	1127	0.451	4	0.00431	4	0.00431
WBC DEC @ 0.995 DTs	235885	480440	226555	0.491	8	0.00003	11	0.00005
WBC DEC @ 0.997 DTs	137637	259718	122081	0.530	8	0.00006	12	0.00009
WBC DEC @ 0.999 DTs	26887	42439	15552	0.634	4	0.00015	13	0.00048
WBC RAN @ 0.995 DTs	1209471	1608117	399646	0.752	9	0.00001	15	0.00001
WBC RAN @ 0.997 DTs	178456	265609	87153	0.672	5	0.00003	8	0.00004
WBC RAN @ 0.999 DTs	7306	13857	6551	0.527	4	0.00055	11	0.00151

With Simple ordering the ratio of cAFS/cTST does not show a material difference for either NNs or DT\S. The cPOF remains small.

Table 33

WBC Using Shapley Ordering

Ordering Shapley	cAFS	cTST	cFST	Ratio cAFS / cTST	POF @ unit cost	Ratio cPOF / cAFS @ unit cost	cPOF @ random cost	Ratio cPOF / cAFS @ random cost
WBC INC @ 0.995 NNs	330	1647	1317	0.200	10	0.03030	6	0.01818
WBC INC @ 0.997 NNs	15	155	140	0.097	2	0.13333	2	0.13333
WBC INC @ 0.999 NNs	21	138	117	0.152	3	0.14286	4	0.19048
WBC DEC @ 0.995 NNs	573	2501	3074	0.229	2	0.00349	3	0.00524
WBC DEC @ 0.997 NNs	15	110	95	0.136	1	0.06667	1	0.06667
WBC DEC @ 0.999 NNs	5	72	67	0.069	1	0.20000	1	0.20000
WBC INC @ 0.995 DTs	135993	250882	114889	0.542	9	0.00007	11	0.00008
WBC INC @ 0.997 DTs	166980	282146	115166	0.592	7	0.00004	11	0.00007
WBC INC @ 0.999 DTs	178	633	455	0.281	3	0.01685	7	0.03933
WBC DEC @ 0.995 DTs	691134	999257	308123	0.692	4	0.00001	10	0.00001
WBC DEC @ 0.997 DTs	28536	52586	24050	0.543	9	0.00032	4	0.00014
WBC DEC @ 0.999 DTs	316	864	548	0.366	4	0.01266	5	0.01582

The ratio of cAFS/cTST is consistently higher for Decreasing than for Increasing with respect to both NNs and DTS. The cPOF remains small, even though the cAFS runs into the hundreds of thousands.

References

Al-Dlaeen, D., & Alashqur, A. (2014). Using decision tree classification to assist in the prediction of Alzheimer's disease. *2014 6th International Conference on Computer Science and Information Technology (CSIT),* 122-126. doi:10.1109/csit.2014.6805989

Aljaaf, A. J., Al-Jumeily, D., Hussain, A. J., Dawson, T., Fergus, P., & Al-Jumaily, M. (2015). Predicting the likelihood of heart failure with a multi level risk assessment using decision tree. *2015 Third International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE),* 101-106. doi:10.1109/taeece.2015.7113608

Almuallim, H., & Dietterich, T. (1991). Learning with many irrelevant features. *AAAI'91 Proceedings of the ninth National conference on Artificial intelligence* ,547-552.

Almuallim , H., & Dietterich, T. (1992). Efficient Algorithms for Identifying Relevant Features. *Proceedings of the Ninth Canadian Conference on Artificial Intelligence,*38-45.

Al-Salihy, N. K., & Ibrikci, T. (2017). Classifying breast cancer by using decision tree algorithms. *Proceedings of the 6th International Conference on Software and Computer Applications - ICSCA 17*. doi:10.1145/3056662.3056716

Arauzo A., Benítez J.M., Castro J.L. (2003) C-FOCUS: A continuous extension of FOCUS. In: Benítez J.M., Cordón O., Hoffmann F., Roy R. (eds) Advances in Soft Computing. Springer, London, doi:10.1007/978-1-4471-3744-3_22

Arista-Jalife, A., & Arista-Viveros, H. A. (2012). Artificial Neural Networks as auxiliary tools in the diagnosis of malnutrition related diseases. *2012 9th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE),* 968-973. doi:10.1109/iceee.2012.6421161

Aruntammanak, W., Aunhathaweesup, Y., Wongseree, W., Leelasantitham, A., & Kiattisin, S. (2013). Diagnose flat foot from foot print image based on neural network. *The 6th 2013 Biomedical Engineering International Conference,* 1-5. doi:10.1109/bmeicon.2013.6687684

Ayeldeen, H., Shaker, O., Ayeldeen, G., & Anwar, K. M. (2015). Prediction of liver fibrosis stages by machine learning model: A decision tree approach. *2015 Third World Conference on Complex Systems (WCCS)*. doi:10.1109/icocs.2015.7483212

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks,5*(4), 537-550. doi:10.1109/72.298224

Bazgir, O., Frounchi, J., Habibi, S. A., Palma, L., & Pierleoni, P. (2015). A neural network system for diagnosis and assessment of tremor in Parkinson disease patients. *2015 22nd Iranian Conference on Biomedical Engineering (ICBME),* 1-5. doi:10.1109/icbme.2015.7404105

Cardie, C. (1993). Using Decision Trees to Improve Case-Based Learning. *Machine Learning Proceedings 1993,*25-32. doi:10.1016/b978-1-55860-307-3.50010-1

Chaddad, A., Zinn, P. O., & Colen, R. R. (2014). Brain tumor identification using Gaussian Mixture Model features and Decision Trees classifier. *2014 48th Annual Conference on Information Sciences and Systems (CISS)*. doi:10.1109/ciss.2014.6814077

Chen, J., & Ngo, C. (2016). Deep-based Ingredient Recognition for Cooking Recipe Retrieval. *Proceedings of the 2016 ACM on Multimedia Conference - MM '16,* 32-41. doi:10.1145/2964284.2964315

Chen, M., Zheng, A., Lloyd, J., Jordan, M., & Brewer, E. (2004). Failure diagnosis using decision trees. *International Conference on Autonomic Computing, 2004. Proceedings.*doi:10.1109/icac.2004.1301345

Cheng, Y., & Wang, P. (2015). Packet Classification Using Dynamically Generated Decision Trees. *IEEE Transactions on Computers, 64*(2), 582-586. doi:10.1109/tc.2013.227

Chunekar, V. N., & Ambulgekar, H. P. (2009). Approach of Neural Network to Diagnose Breast Cancer on three different Data Set. *2009 International Conference on Advances in Recent Technologies in Communication and Computing*, 893-895. doi:10.1109/artcom.2009.225

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing. *Proceedings of the 25th international conference on Machine learning - ICML '08*, 160-167. doi:10.1145/1390156.1390177

Coppini, G., Miniati, M., Paterni, M., Monti, S., & Ferdeghini, E. (2007). Computer-aided diagnosis of emphysema in COPD patients: Neural-network-based analysis of lung shape in digital chest radiographs. *Medical Engineering & Physics*, *29(1)*, 76–86. doi: 10.1016/j.medengphy.2006.02.001

Covington, P., Adams, J., & Sargin, E. (2016). Deep Neural Networks for YouTube Recommendations. *Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16*, 191-198. doi:10.1145/2959100.2959190

Cui, Y., Xiong, H., Zheng, K., & Chen, J. (2012). On the application of BP neural network based on Levenberg-Marquardt algorithm in the diagnosis of mental disorders. *2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, 1940-1943. doi:10.1109/cecnet.2012.6201941

Das, S. (2001). Filters, Wrappers and a Boosting-Based Hybrid for Feature Selection. *Proceeding ICML '01 Proceedings of the Eighteenth International Conference on Machine Learning,*74-81.

Dhakate, M., & Ingole, A. (2015). Diagnosis of pomegranate plant diseases using neural network. *2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, 1-4. doi:10.1109/ncvpripg.2015.7490056

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. New York, NY: John Wiley & Sons.

El-Solh, A., Hsiao, C., Goodnough, S., Serghani, J., & Grant, B. (1999). Predicting Active Pulmonary Tuberculosis Using an Artificial Neural Network. *Chest, 116*(4), 968-973. doi:10.1378/chest.116.4.968

Elveren, E., & Yumuşak, N. (2009). Tuberculosis Disease Diagnosis Using Artificial Neural Network Trained with Genetic Algorithm. *Journal of Medical Systems, 35(3)*, 329-332. doi:10.1007/s10916-009-9369-3

Er, O., Yumusak, N., & Temurtas, F. (2010). Chest diseases diagnosis using artificial neural networks. *Expert Systems with Applications, 37(12)*, 7648-7655. doi:10.1016/j.eswa.2010.04.078

Er, O., Sertkaya, C., Temurtas, F., & Tanrikulu, A. (2009). A comparative study on chronic obstructive pulmonary and pneumonia diseases diagnosis using neural networks and artificial immune system. *Journal of Medical Systems, 33*, 485-492. doi: 10.1007/s10916-008-9209-x

Fleuret, F. (2004). Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research,5*, 1531-1555.

Floares, A., & Birlutiu, A. (2012). Decision tree models for developing molecular classifiers for cancer diagnosis. *The 2012 International Joint Conference on Neural Networks (IJCNN)*. doi:10.1109/ijcnn.2012.6252781

Fong, S., Zhuang, Y., & He, J. (2012). Not every friend on a social network can be trusted: Classifying imposters using decision trees. *The First International Conference on Future Generation Communication Technologies,* 58-63. doi:10.1109/fgct.2012.6476584

Gao, Z., Chin, C. S., Woo, W. L., Jia, J., & Toh, W. D. (2015). Genetic Algorithm based Back-Propagation Neural Network approach for fault diagnosis in lithium-ion battery system. *2015 6th International Conference on Power Electronics Systems and Applications (PESA)*, 1-6. doi:10.1109/pesa.2015.7398911

Greiner, R., Grove, A. J., & Roth, D. (2002). Learning cost-sensitive active classifiers☆☆This extends the short conference paper [19]. *Artificial Intelligence,139*(2), 137-174. doi:10.1016/s0004-3702(02)00209-6

Gu, J., Deng, C., Lin, X., & Yu, D. (2012). Expert system for fish disease diagnosis based on fuzzy neural network. *2012 Third International Conference on Intelligent Control and Information Processing*, 146-149. doi:10.1109/icicip.2012.6391445

Guo, Q., & Zhang, M. (2008). A novel approach for fault diagnosis of steam turbine based on neural network and genetic algorithm. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 25-29. doi:10.1109/ijcnn.2008.4633762

Guresen, E., Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert Systems with Applications, 38(8)*, 10389-10397. doi:10.1016/j.eswa.2011.02.068

Gutierrez, A. (2015). Influence of Wavelets and Boundary Conditions on the Diagnosis of Multiple Sclerosis Using Artificial Neural Networks. *2015 Annual Global Online Conference on Information and Computer Technology (GOCICT)*, 6-10. doi:10.1109/gocict.2015.10

He, M., Zhang, J., & Vittal, V. (2013). Robust Online Dynamic Security Assessment Using Adaptive Ensemble Decision-Tree Learning. *IEEE Transactions on Power Systems,28*(4), 4089-4098. doi:10.1109/tpwrs.2013.2266617

He, X., Cai, D., & Niyogi, P. (2005). Laplacian score for feature selection. *Proceeding NIPS'05 Proceedings of the 18th International Conference on Neural Information Processing Systems,*507-514.

Huang, C., Yan, B., Jiang, H., & Wang, D. (2008). Combining Voxel-based Morphometry with Artificial Neural Network Theory in the Application Research of Diagnosing Alzheimer's Disease. 2008 *International Conference on BioMedical Engineering and Informatics*, 250-254. doi:10.1109/bmei.2008.245

Ibrahim, A. O., Shamsuddin, S. M., Saleh, A. Y., Abdelmaboud, A., & Ali, A. (2015). Intelligent multi-objective classifier for breast cancer diagnosis based on multilayer perceptron neural network and Differential Evolution. *2015 International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE)*, 422-427. doi:10.1109/iccneee.2015.7381405

Jakulin, A. (2005) *Machine learning based on attribute interactions* (Unpublished doctoral dissertation). University of Ljubljana, Sezana.

Ji, J., Jiang, H., Zhao, B., & Zhai, P. (2015). Crucial Data Selection Based on Random Weight Neural Network. *2015 IEEE International Conference on Systems, Man, and Cybernetics*, 1017-1022. doi:10.1109/smc.2015.184

John, G. H., Kohavi, R., & Pfleger, K. (1994). Irrelevant Features and the Subset Selection Problem. *Machine Learning Proceedings 1994,*121-129. doi:10.1016/b978-1-55860-335-6.50023-4

Jyothi, S., & Vanisree, K. (2016). Congenital Heart Septum Defect Diagnosis on Chest X-Ray Features Using Neural Networks. *2016 Second International Conference on Computational Intelligence & Communication Technology (CICT)*, 265-269. doi:10.1109/cict.2016.59

Kabari, L. G., & Bakpo, F. S. (2009). Diagnosing skin diseases using an artificial neural network. *2009 2nd International Conference on Adaptive Science & Technology (ICAST)*, 187-191. doi:10.1109/icastech.2009.5409725

Kadi, I., & Idri, A. (2015). A Decision Tree-Based Approach for Cardiovascular Dysautonomias Diagnosis: A Case Study. *2015 IEEE Symposium Series on Computational Intelligence*. doi:10.1109/ssci.2015.121

Kelleher, J. D., Namee, B. M., & DArcy, A. (2015). *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. Massachusetts: The MIT Press.

Kira, K. & Rendell, L. A. (1992). The Feature Selection Problem: Traditional Methods and a New Algorithm.. In W. R. Swartout (ed.), *AAAI* (p./pp. 129-134), : AAAI Press / The MIT Press. ISBN: 0-262-51063-4

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence, 97*(1-2), 273-324. doi:10.1016/s0004-3702(97)00043-x

Kondo, T., Ueno, J., & Takao, S. (2012). Hybrid multi-layered GMDH-type neural network self-selecting various neurons and its application to medical image diagnosis of liver cancer. *The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems*, 1919-1924. doi:10.1109/scis-isis.2012.6505292

Kononenko, I. (1994). Estimating attributes: Analysis and extensions of RELIEF. *Machine Learning: ECML-94 Lecture Notes in Computer Science,*171-182. doi:10.1007/3-540-57868-4_57

Kumar, A., Hanmandlu, M., Das, A., & Gupta, H. M. (2012). Biometric based personal authentication using fuzzy binary decision tree. *2012 5th IAPR International Conference on Biometrics (ICB)*. doi:10.1109/icb.2012.6199783

Kumar, D., & Krishniah, V. (2016). An automated framework for stroke and hemorrhage detection using decision tree classifier. *2016 International Conference on Communication and Electronics Systems (ICCES)*. doi:10.1109/cesys.2016.7889861

Kumar, M., Sharma, A., & Agarwal, S. (2014). Clinical decision support system for diabetes disease diagnosis using optimized neural network. *2014 Students Conference on Engineering and Systems*, 201-206. doi:10.1109/sces.2014.6880051

Lakshmi, B., Indumathi, T., & Ravi, N. (2015). A novel health monitoring approach for pregnant women. *2015 International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*. doi:10.1109/erect.2015.7499035

Lee, S., Park, J., & Kang, K. (2015). Assessing wine quality using a decision tree. *2015 IEEE International Symposium on Systems Engineering (ISSE)*. doi:10.1109/syseng.2015.7302752

Lewis, D. (1992). Feature selection and feature extraction for text categorization. *Proceedings of the workshop on Speech and Natural Language - HLT 91*. doi:10.3115/1075527.1075574

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: a data perspective. *ACM Computing Surveys,50*(6), 94:1-94:45. doi:10.1145/3136625

Li, Y., Chen, C., & Wasserman, W. W. (2015). Deep Feature Selection: Theory and Application to Identify Enhancers and Promoters. *Lecture Notes in Computer Science Research in Computational Molecular Biology,*205-217. doi:10.1007/978-3-319-16706-0_20

Lin, C., Hsieh, S., & Hu, Y. (2013). Fuzzy Neural Network-Based Influenza Diagnostic System. *2013 First International Symposium on Computing and Networking*, 633-635. doi:10.1109/candar.2013.115

Lin, D., & Tang, X. (2006). Conditional Infomax Learning: An Integrated Framework for Feature Extraction and Fusion. *Computer Vision – ECCV 2006 Lecture Notes in Computer Science,*68-82. doi:10.1007/11744023_6

Liu, M., & Dong, X. (2012). The application of improved BP neural network in the diagnosis of breast tumors. *2012 International Conference on Systems and Informatics (ICSAI2012)*, 1239-1242. doi:10.1109/icsai.2012.6223260

Long, X., & Wu, Y. (2012). Application of Decision Tree in Student Achievement Evaluation. *2012 International Conference on Computer Science and Electronics Engineering,* 243-247. doi:10.1109/iccsee.2012.169

Luculescu, M. C., & Lache, S. (2008). Using artificial neural networks in a Computer Aided Diagnosis system for Macular diseases. *2008 IEEE International Conference on Automation, Quality and Testing, Robotics*, 143-148. doi:10.1109/aqtr.2008.4588899

Maity, M., Sarkar, P., & Chakraborty, C. (2012). Computer-assisted approach to anemic erythrocyte classification using blood pathological information. *2012 Third International Conference on Emerging Applications of Information Technology,*116-121. doi:10.1109/eait.2012.6407875

Marcos, J. V., Hornero, R., Alvarez, D., Campo, F. D., & Lopez, M. (2007). Applying Neural Network Classifiers in the Diagnosis of the Obstructive Sleep Apnea Syndrome from Nocturnal Pulse Oximetric Recordings. *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. doi:10.1109/iembs.2007.4353507

Masaeli, M., Yan, Y., Cui, Y., Fung, G., & Dy, J. G. (2010). Convex Principal Feature Selection. *Proceedings of the 2010 SIAM International Conference on Data Mining,*619-628. doi:10.1137/1.9781611972801.54

Meyer, P. E., & Bontempi, G. (2006). On the Use of Variable Complementarity for Feature Selection in Cancer Classification. *Lecture Notes in Computer Science Applications of Evolutionary Computing,*91-102. doi:10.1007/11732242_9

Meyer, P. E., Schretter, C., & Bontempi, G. (2008). Information-Theoretic Feature Selection in Microarray Data Using Variable Complementarity. *IEEE Journal of Selected Topics in Signal Processing,2*(3), 261-274. doi:10.1109/jstsp.2008.923858

Nie, F., Xiang, S., Jia, Y., Zhang, C., Yan, S., (2008), Trace ratio criterion for feature selection, A. Cohn (Ed.), *AAAI'08 Proceedings of the 23rd national conference on Artificial intelligence - Volume 2*, Chicago, Illinois, AAAI Press.

Ochotorena, C. N., Yap, C. A., Dadios, E., & Sybingco, E. (2012). Robust stock trading using fuzzy decision trees. *2012 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*. doi:10.1109/cifer.2012.6327785

Oh, J., Kim, T., & Hong, H. (2013). Using Binary Decision Tree and Multiclass SVM for Human Gesture Recognition. *2013 International Conference on Information Science and Applications (ICISA)*. doi:10.1109/icisa.2013.6579388

Palfy, M., & Papez, B. J. (2007). Diagnosis of Carpal Tunnel Syndrome from Thermal Images Using Artificial Neural Networks. Twentieth *IEEE International Symposium on Computer-Based Medical Systems (CBMS'07)*, 59-64. doi:10.1109/cbms.2007.40

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence,27*(8), 1226-1238. doi:10.1109/tpami.2005.159

Pytel, K., Nawarycz, T., Drygas, W., & Ostrowska-Nawarycz, L. (2015). Anthropometric Predictors and Artificial Neural Networks in the diagnosis of Hypertension. *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems*, 287-290. doi:10.15439/2015f246

Sammouda, R. S., Wang, X., & Basilion, J. P. (2015). Hopfield Neural Network for the segmentation of Near Infrared Fluorescent images for diagnosing prostate cancer. *2015 6th International Conference on Information and Communication Systems (ICICS)*, 111-118. doi:10.1109/iacs.2015.7103212

Santoro, D. M., Nicoletti, M. D., & Hruschka, E. R. (2007). C-Focus-3: a C-Focus with a New Heuristic Search Strategy. *Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)*. doi:10.1109/isda.2007.20

Seo, J., Yu, J., Lee, J., & Choi, K. (2016). A new approach to binarizing neural networks. *2016 International SoC Design Conference (ISOCC)*, 77-78. doi:10.1109/isocc.2016.7799741

Sharma, M. (2014). Z - CRIME: A data mining tool for the detection of suspicious criminal activities based on decision tree. *2014 International Conference on Data Mining and Intelligent Computing (ICDMIC)*. doi:10.1109/icdmic.2014.6954268

Shroff, S., Pise, S., Chalekar, P., & Panicker, S. S. (2015). Thyroid disease diagnosis: A survey. *2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO)*. doi:10.1109/isco.2015.7282384

Shukla, A., Tiwari, R., Kaur, P., & Janghel, R. (2009). Diagnosis of Thyroid Disorders using Artificial Neural Networks. *2009 IEEE International Advance Computing Conference*, 1016-1020. doi:10.1109/iadcc.2009.4809154

Songthung, P., & Sripanidkulchai, K. (2016). Improving type 2 diabetes mellitus risk prediction using classification. *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. doi:10.1109/jcsse.2016.7748866

Stein, G., Chen, B., Wu, A. S., & Hua, K. A. (2005). Decision tree classifier for network intrusion detection with GA-based feature selection. *Proceedings of the 43rd annual southeast regional conference on - ACM-SE 43,*2-136-2-141. doi:10.1145/1167253.1167288

Sun, Y., Lu, J., Kobayashi, A., & Yahagi, T. (2005). Neural Network Ultrsonographic Diagnosis System of Cirrhosis Using DWT for Preprocessing. *2005 IEEE International Symposium on Circuits and Systems*, 2783-2786. doi:10.1109/iscas.2005.1465204

Syiam, M. (1994). A neural network expert system for diagnosing eye diseases. *Proceedings of the Tenth Conference on Artificial Intelligence for Applications*, 491-492. doi:10.1109/caia.1994.323624

Tan, T., Quek, C., & Ng, G. (2005). Ovarian cancer diagnosis using complementary learning fuzzy neural network. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks*, 2005., 3034-3039. doi:10.1109/ijcnn.2005.1556409

Tartar, A., Kilic, N., & Akan, A. (2013). A new method for pulmonary nodule detection using decision trees. *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. doi:10.1109/embc.2013.6611257

Thakur, A., Guleria, P., & Bansal, N. (2016). Symptom & risk factor based diagnosis of Gum diseases using neural network. *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*, 101-104. doi:10.1109/confluence.2016.7508095

Venu, M., Kiran, R. U., & Kiranmai, R. (2012). A robust neural network classifier to model the compressive strength of high performance concrete using feature subset selection. *Proceedings of the 5th ACM COMPUTE Conference on Intelligent & scalable system technologies - COMPUTE '12*, 1-8. doi:10.1145/2459118.2459119

Vidal-Naquet, M., & Ullman, S. (2003). Object recognition with informative features and linear classification. *Proceedings Ninth IEEE International Conference on Computer Vision*. doi:10.1109/iccv.2003.1238356

Vijayasarveswari, V., Khatun, S., Jusoh, M., Fakir, M., & Ali, M. S., (2016). Early breast health screening performance verification based on UWB imaging and neural network. *2016 3rd International Conference on Electronic Design (ICED)*, 517-521. doi:10.1109/iced.2016.7804699

Xiang, Y., Tian, J., Zhang, Z., & Dai, Y. (2009). Diagnosis of Endometrial Cancer Based on Back-Propagation Neural Network and Near-Infrared Spectroscopy of Tissue. *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, 508-512. doi:10.1109/fskd.2009.470

Yalcin, H., & Razavi, S. (2016). Plant classification using convolutional neural networks. 2016 *Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*. 1 - 5. doi:10.1109/agro-geoinformatics.2016.7577698

Yan, Z., Zhang, H., Wang, B., Paris, S., & Yu, Y. (2016). Automatic Photo Adjustment Using Deep Neural Networks. *ACM Transactions on Graphics, 35(2)*, 1-15. doi:10.1145/2790296

Yu, L., & Liu, H. (2003). Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution. In T. Fawcett & N. Mishra (Eds.) *Proceedings of the Twentieth International Conference on Machine Learning,*(pp. 856-863). Washington, DC, United States.

Yuncu, E., Hacihabiboglu, H., & Bozsahin, C. (2014). Automatic Speech Emotion Recognition Using Auditory Models with Binary Decision Tree and SVM. *2014 22nd International Conference on Pattern Recognition*. doi:10.1109/icpr.2014.143

Zhao, Z., & Liu, H. (2007). Spectral feature selection for supervised and unsupervised learning. *Proceedings of the 24th international conference on Machine learning - ICML 07*. doi:10.1145/1273496.1273641

Zhang, T., Tang, W., & Sazonov, E. S. (2012). Classification of posture and activities by using decision trees. *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society,* 4353-4356. doi:10.1109/embc.2012.6346930

Zhou, J., Liu, J., Narayan, V. A., & Ye, J. (2012). Modeling disease progression via fused sparse group lasso. *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 12*. doi:10.1145/2339530.2339702

Zou, K., Sun, W., Yu, H., & Liu, F. (2012). ID3 Decision Tree in Fraud Detection Application. *2012 International Conference on Computer Science and Electronics Engineering,* 399-402. doi:10.1109/iccsee.2012.24